

**PENERAPAN ALGORITMA GENETIKA DAN PERBANDINGANNYA
DENGAN ALGORITMA GREEDY DALAM PENYELESAIAN KNAPSACK
PROBLEM**



SKRIPSI

Diajukan untuk memenuhi salah satu syarat guna mencapai gelar

Sarjana Komputer pada Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

UIN Alauddin Makassar

Oleh:

NUR AIMA ALI

NIM : 60200111067

FAKULTAS SAINS DAN TEKNOLOGI

UIN ALAUDDIN MAKASSAR

2017

PERSETUJUAN PEMBIMBING

Pembimbing penulisan skripsi saudara **Nur Aima Ali : 60200111067**, mahasiswa Jurusan Teknik Informatika pada Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar, setelah dengan seksama meneliti dan mengoreksi skripsi yang bersangkutan dengan judul, “Penerapan Algoritma Genetika dan Perbandingannya dengan Algoritma *Greedy* dalam Penyelesaian *Knapsack Problem*”, memandang bahwa skripsi tersebut telah memenuhi syarat-syarat ilmiah dan dapat disetujui untuk diajukan ke sidang Munaqasyah.

Demikian persetujuan ini diberikan untuk proses selanjutnya.

Makassar, Agustus 2017

Pembimbing I



Faisal Akib, S.Kom., M.Kom.
NIP. 19761212 200501 1 005

Pembimbing II



Nur Afif, S.T., M.T.
NIP. 19810224 200912 1 003

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Nur Aima Ali
NIM : 60200111067
Tempat/Tgl. Lahir : Watampone, 14 Juli 1992
Jurusan : Teknik Informatika
Fakultas/Program : Sains dan Teknologi
Judul : Penerapan Algoritma Genetika dan Perbandingannya
dengan Algoritma *Greedy* dalam Penyelesaian
Knapsack Problem

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar merupakan hasil karya saya sendiri. Jika di kemudian hari terbukti bahwa ini merupakan duplikasi, tiruan, plagiat, atau dibuat oleh orang lain, sebagian atau seluruhnya, maka skripsi dan gelar yang diperoleh karenanya batal demi hukum.

Makassar, Agustus 2017
Penyusun,

Nur Aima Ali
NIM : 60200111067

PENGESAHAN SKRIPSI

Skripsi yang berjudul “Penerapan Algoritma Genetika dan Perbandingannya dengan Algoritma *Greedy* dalam Penyelesaian *Knapsack Problem*” yang disusun oleh Nur Aima Ali, NIM 60200111067, mahasiswa Jurusan Teknik Informatika pada Fakultas Sains dan Teknologi UIN Alauddin Makassar, telah diuji dan dipertahankan dalam sidang munaqasyah yang diselenggarakan pada hari Jumat 18 Agustus 2017 M dinyatakan telah dapat diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana dalam Ilmu Teknik Informatika, Jurusan Teknik Informatika.

Makassar, Agustus 2017

DEWAN PENGUJI:

Ketua	: Dr. Muh. Thahir Maloko, M.HI.	(.....)
Sekretaris	: A. Hutami Endang, S.Kom., M.Kom	(.....)
Munaqisy I	: Dr. H. Kamaruddin Tone, M.M.	(.....)
Munaqisy II	: Faisal, S.T., M.T.	(.....)
Munaqisy III	: Dr. Anwar Sadat, M.Ag.	(.....)
Pembimbing I	: Faisal Akib, S.Kom., M.Kom.	(.....)
Pembimbing II	: Nur Afif, S.T., M.T.	(.....)

Diketahui oleh:



Dekan Fakultas Sains dan Teknologi
UIN Alauddin Makassar,

Prof. Dr. H. Arifuddin, M.Ag.
NIP. 19691205 199303 1 001

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah, puji syukur kehadiran Allah SWT atas rahmat, hidayah, pertolongan-Nya sehingga penulis dapat menyusun dan menyelesaikan laporan skripsi dengan judul “Penerapan Algoritma Genetika dan Perbandingannya dengan Algoritma *Greedy* dalam Penyelesaian *Knapsack Problem*”. Skripsi ini disusun untuk memenuhi syarat dalam menyelesaikan studi dan memperoleh gelar sarjana untuk program studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Alauddin Makassar.

Dalam pelaksanaan penelitian sampai pembuatan skripsi ini, penulis banyak sekali mengalami kesulitan dan hambatan. Tetapi berkat keteguhan dan kesabaran penulis akhirnya skripsi ini dapat diselesaikan juga. Hal ini karena dukungan dan bantuan dari berbagai pihak yang dengan senang hati memberikan dorongan dan bimbingan yang tak henti-hentinya kepada penulis. Terima kasih yang tak terhingga kepada abba dan ummi tercinta Drs. Muhammad Ali Musa, M.M dan Dra. Aryani yang selalu memberikan doa, kasih sayang, dan dukungan baik moral maupun materil. Tak akan pernah cukup kata untuk mengungkapkan rasa terima kasih ananda buat abba dan ummi tercinta. Dan kepada adik-adik penulis Ishaq, Atira, Ikram, Ikhsan, Ikhwan, Ilman, dan Idham yang juga selalu setia menyemati dan mendoakan agar terselesainya skripsi penulis.

Melalui kesempatan ini, penulis menyampaikan rasa terima kasih yang sebesar-besarnya dan penghargaan yang setinggi-tingginya kepada :

1. Prof. Dr. H. Musafir Pababbari, M.Si. selaku Rektor Universitas Islam Negeri (UIN) Alauddin Makassar.
2. Prof. Dr. H. Arifuddin, M.Ag. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar.
3. Ketua Jurusan Teknik Informatika, Faisal, S.T., M.T. sekaligus penguji II dan Sekretaris Jurusan Teknik Informatika Andi Muhammad Syafar, S.T., M.T.
4. Faisal Akib, S.Kom., M.Kom. selaku pembimbing I dan Nur Afif, S.T., M.T. selaku pembimbing II yang telah membimbing dan membantu penulis untuk mengembangkan pemikiran dalam penyusunan skripsi ini hingga selesai.
5. Dr. H. Kamaruddin Tone, M.M. selaku penguji I dan Dr. Anwar Sadat, M.Ag. selaku penguji III yang telah menguji dan membimbing dalam penulisan skripsi ini.
6. Seluruh dosen, staf dan karyawan Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Alauddin Makassar yang telah banyak memberikan sumbangsih baik tenaga maupun pikiran.
7. Pihak pengelola jasa peti kemas kawasan logistik terpadu Makassar yang telah membantu memberikan informasi data usaha pengelolaan peti kemas yang ada di Makassar dan telah memberikan izin untuk melakukan penelitian.
8. Sahabat-sahabat penulis yang tidak henti-hentinya memberikan dukungan dan bantuan selama penulisan skripsi ini. Terima kasih kesayanganku Ulfa

Muthmainnah, Nurul Mukhlisah, Nur Wahyuni Ulfa MS, Nurhikmah Arifin, Nurfadila, Susanti, Putri Pahrurnisa, Vestiana Aza, dan Nur Ulfaida Nasmar.

9. Teman-teman Teknik Informatika angkatan 2011 yang telah menjadi saudara seperjuangan menjalani suka dan duka bersama dalam menempuh pendidikan di kampus.
10. Seluruh pihak yang tidak dapat penulis sebutkan satu per satu, namun telah banyak terlibat membantu penulis dalam proses penyusunan skripsi ini.

Semoga skripsi ini dapat berguna bagi para pembaca sekalian. Lebih dan kurangnya penulis mohon maaf yang sebesar-besarnya, semoga Allah swt. melimpahkan rahmat-Nya kepada kita semua. Aamiin.

Makassar, Juli 2017

Penyusun,

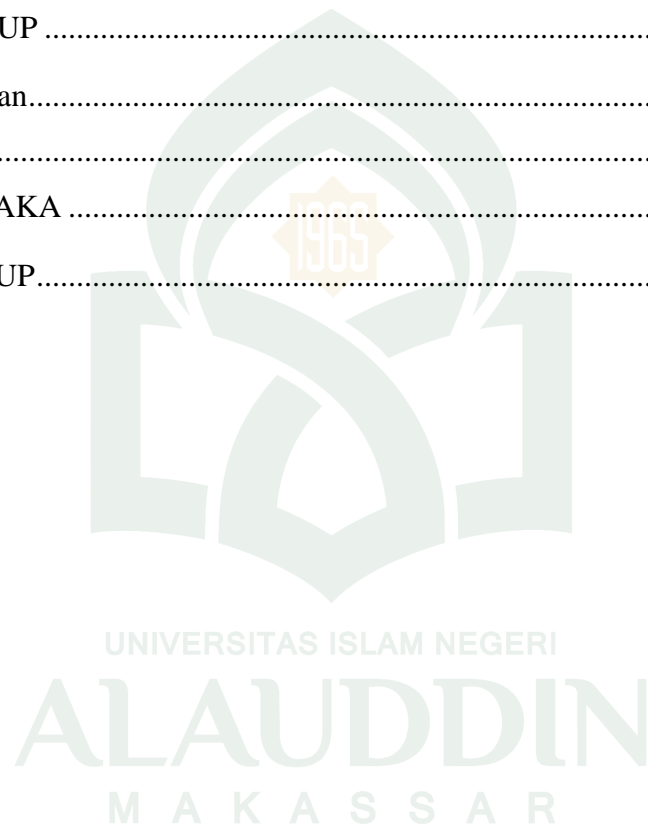
UNIVERSITAS ISLAM NEGERI
ALA UDDIN
M A K A S S A R
Nur Aima Ali
NIM : 60200111067

DAFTAR ISI

HALAMAN JUDUL.....	i
PERSETUJUAN PEMBIMBING.....	ii
PERNYATAAN KEASLIAN SKRIPSI.....	iii
PENGESAHAN SKRIPSI	iv
KATA PENGANTAR	v
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xiii
ABSTRAK	xiv
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah.....	1
B. Rumusan Masalah	6
C. Fokus Penelitian dan Deskripsi Fokus	7
D. Kajian Pustaka.....	10
E. Tujuan dan Kegunaan Penelitian.....	12
1. Tujuan Penelitian.....	12
2. Kegunaan Penelitian.....	12
BAB II TINJAUAN TEORITIS	14
A. Tinjauan Al-Quran dan Hadist	14
B. Aplikasi	16
C. Optimasi	17
D. Peti Kemas.....	18
E. Knapsack Problem.....	18

F.	Algoritma Genetika	19
G.	Algoritma Greedy.....	22
H.	Web	23
I.	Hypertext Markup Language (HTML)	24
J.	Hypertext Preprocessor (PHP)	24
K.	MySQL.....	25
L.	Cascading Style Sheet (CSS)	25
M.	Javascript.....	26
N.	Daftar Simbol	27
1.	Flowmap.....	27
2.	Flowchart.....	28
BAB III METODE PENELITIAN		31
A.	Jenis Penelitian.....	31
B.	Pendekatan Penelitian	31
C.	Sumber Data.....	31
D.	Metode Pengumpulan Data	31
1.	Observasi.....	32
2.	Studi Literatur	32
E.	Instrumen Penelitian.....	32
1.	Perangkat Keras.....	32
2.	Perangkat Lunak.....	33
F.	Teknik Pengolahan dan Analisis Data	33
G.	Metode Perancangan Aplikasi.....	34
H.	Teknik Pengujian Sistem.....	36
BAB IV ANALISIS DAN PERANCANGAN SISTEM		38
A.	Analisis Sistem yang Sedang Berjalan.....	38
B.	Analisis Sistem yang Diusulkan.....	39
1.	Analisis Masalah	39

C. Perancangan Sistem.....	40
1. Perancangan Interface Aplikasi.....	40
2. Flowchart.....	44
BAB V IMPLEMENTASI DAN ANALISIS HASIL	47
A. Implementasi	47
B. Pengujian Sistem	51
BAB VI PENUTUP	67
A. Kesimpulan.....	67
B. Saran.....	68
DAFTAR PUSTAKA	xv
RIWAYAT HIDUP.....	xvii



DAFTAR GAMBAR

Gambar III. 1 Model <i>Waterfall</i>	33
Gambar IV. 1 <i>Flowmap</i> Analisis yang sedang berjalan.....	37
Gambar IV. 2 Desain <i>Interface</i> Halaman Utama	40
Gambar IV. 3 Desain <i>Interface</i> Input Jumlah Barang.....	40
Gambar IV. 4 Desain <i>Interface</i> Input Nama, Berat, Harga Barang dan Kapasitas	41
Gambar IV. 5 Desain <i>Interface</i> Hasil Genetika	41
Gambar IV. 6 Desain <i>Interface</i> Hasil <i>Greedy</i>	42
Gambar IV. 7 Desain <i>Interface</i> Hasil Kombinasi Barang Terpilih Genetika	42
Gambar IV. 8 Desain <i>Interface</i> Hasil Kombinasi Barang Terpilih <i>Greedy</i>	43
Gambar IV. 9 Desain <i>Flowchart</i> Algoritma Genetika	44
Gambar IV. 10 Desain <i>Flowchart</i> Algoritma <i>Greedy</i>	45
Gambar V. 1 <i>Interface</i> Halaman Utama.....	46
Gambar V. 2 <i>Interface</i> Input Jumlah Barang	47
Gambar V. 3 <i>Interface</i> Input Nama, Berat, Harga Barang dan Kapasitas <i>Greedy</i> <i>Algorithm</i>	47
Gambar V. 4 <i>Interface</i> Hasil Kombinasi Barang Terpilih <i>Greedy Algorithm</i>	48
Gambar V.5 <i>Interface</i> Input Jumlah Barang dengan Genetic Algorithm	49
Gambar V.6 <i>Interface</i> Input Nama, Harga, Berat Barang dan Kapasitas dengan <i>Genetic Algorithm</i>	49

Gambar V. 7 <i>Interface</i> Hasil Kombinasi Barang Terpilih <i>Genetic Algorithm</i>	50
Gambar V. 8 Pengujian Whitebox Algoritma <i>Greedy</i>	52
Gambar V. 9 Pengujian Whitebox Algoritma Genetika	57



DAFTAR TABEL

Tabel II.1 Daftar Simbol <i>Flowmap</i> Diagram (Jogiyanto, 2001)	27
Tabel II.2 Daftar Simbol <i>Flowchart</i> (Kristanto, 2003)	29
Tabel III.1 Rancangan Tabel Pengujian <i>Whitebox Testing</i>	37
Tabel III.2 Rancangan Tabel Pengujian Perbandingan Algoritma	25
Tabel V.1 Pengujian <i>Whitebox</i> Algoritma <i>Greedy</i>	54
Tabel V.2 Pengujian <i>Whitebox</i> Algoritma Genetika.....	59
Tabel V.3 Rekapitulasi Hasil Pengujian Perangkat Lunak.....	62
Tabel V.4 Data Ujicoba Barang.....	63
Tabel V.5 Tabel Perbandingan Algoritma <i>Greedy</i> dan Algoritma Genetika Untuk 20 Barang dengan Kapasitas 1000 Kg.....	64
Tabel V.6 Tabel Perbandingan Algoritma <i>Greedy</i> dan Algoritma Genetika Untuk 30 Barang dengan Kapasitas 1100 Kg.....	65
Tabel V.7 Tabel Perbandingan Algoritma <i>Greedy</i> dan Algoritma Genetika Untuk 50 Barang dengan Kapasitas 1300 Kg.....	65

ABSTRAK

Nama : Nur Aima Ali
Nim : 60200111067
Jurusan : Teknik Informatika
Judul : Penerapan Algoritma Genetika dan Perbandingannya dengan Algoritma *Greedy* dalam Penyelesaian *Knapsack Problem*
Pembimbing I : Faisal Akib, S.Kom., M.Kom.
Pembimbing II : Nur Afif, S.T., M.T.

Permasalahan optimasi adalah suatu permasalahan yang mempunyai banyak solusi dan harus bisa ditentukan solusi mana yang dikatakan optimal. Optimasi bermanfaat dalam bidang jasa pengangkutan barang seperti pengangkutan barang ke dalam peti kemas. Dalam usaha tersebut, diinginkan keuntungan yang maksimal untuk mengangkut barang yang ada dengan tidak melebihi batas kapasitas. Permasalahan semacam ini sering dianalogikan dengan permasalahan *Knapsack*. Solusi dari *Knapsack Problem* dalam pengangkutan barang ke dalam peti kemas, dapat dilakukan dengan menerapkan algoritma genetika dan algoritma *greedy*. Algoritma genetika dan *greedy* yang diterapkan ke dalam suatu aplikasi dapat digunakan untuk menyelesaikan permasalahan *Knapsack* pada pengangkutan barang dalam peti kemas dengan perolehan profit lebih besar. Tujuan dari penelitian ini adalah menerapkan algoritma genetika dan membandingkan dengan algoritma *greedy* dalam menyelesaikan *knapsack problem*.

Dalam melakukan penelitian ini, jenis penelitian yang digunakan adalah *design and creation*, metode perancangan yang digunakan adalah metode *waterfall*. Aplikasi ini berbasis web. Pemodelannya menggunakan *flowchart* dan diuji dengan metode pengujian *whitebox*. Hasil dari pengujian *whitebox* berdasarkan perhitungan yang dilakukan dengan rumus *Cyclomatic Complexity* pada *flowgraph* sudah sesuai dengan alur program yang dirancang. Kesimpulan dari penelitian ini adalah baik algoritma genetika dan algoritma *greedy* sama-sama dapat menyelesaikan *knapsack problem*. Akan tetapi algoritma genetika lebih optimal dalam kasus dengan jumlah banyak barang. Sedangkan algoritma *greedy* unggul dalam kompleksitas waktu pencarian solusi.

Kata Kunci : Optimasi, *Knapsack Problem*, Algoritma Genetika, Algoritma *Greedy*

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Peti kemas (*container*) adalah suatu kotak besar berbentuk empat persegi panjang, terbuat dari bahan campuran baja dan tembaga atau bahan lainnya yang tahan terhadap cuaca. Dapat pula disebut gudang kecil yang berjalan untuk mengangkut barang dari satu tempat ke tempat lain, dengan menggunakan alat pengangkutnya seperti truk, kapal angkut atau kereta api sampai ke tempat yang dituju. Peti kemas dirancang secara khusus dengan ukuran tertentu, dapat dipakai berulang kali, dipergunakan untuk menyimpan dan mengangkut muatan dengan jumlah besar dan dapat dipisahkan dari sarana pengangkutnya dengan mudah tanpa harus mengeluarkan isinya. Jasa pengangkutan peti kemas melakukan pengiriman barang antar pulau maupun negara. Adapun jenis barang disesuaikan dengan jenis peti kemas atau *container* yang digunakan dalam pengiriman barang. Hal inilah yang menyebabkan peralihan angkutan barang umum menjadi angkutan barang menggunakan *container*. (Ambarwari dan Yanto, 2010)

Saat ini pengelola jasa peti kemas melakukan pengiriman barang dengan memperhatikan tujuan pengiriman dan jenis barang yang akan dikirim. Sementara perhitungannya dilakukan secara konvensional dengan menghitung jumlah *space* yang digunakan dalam kubik dikali harga perkubiknya. Hasil yang didapat itulah tarif atau biaya pengangkutan barang yang harus dibayar oleh pengguna jasa. Pengiriman barang

ini tidak hanya terjadi sekali saja, tetapi bisa berulang-ulang sehingga dibutuhkan waktu yang lama dalam memproses satu permintaan pengguna jasa pengangkutan.

Dalam menjalankan proses bisnisnya, salah satu kendala yang dihadapi oleh perusahaan peti kemas adalah kurangnya kontrol dalam melakukan muat barang. Setiap menerima barang dari konsumen atau dalam hal ini perusahaan ekspedisi, staf perusahaan peti kemas hanya mencatat data pengirim, apa isi atau komoditinya, dan tujuan pengiriman tanpa memperhitungkan kapasitas dari kendaraan yang akan mengirim. Masalah terjadi saat melakukan muat barang sebanyak mungkin ke dalam *container*, asal angkut barang terjadi hingga terkadang melebihi kapasitas muatan. Apabila *container* sudah penuh, maka barang akan dikirim menggunakan *container* lain di hari berikutnya.

Selain pengangkutan, efisiensi dan keuntungan juga harus dipikirkan, sehingga dibutuhkan keterampilan dalam penyusunan barang agar kapasitas ruang bisa optimal tanpa melebihi kemampuan daya tampung. Dengan pertimbangan tersebut diharapkan dapat diperoleh keuntungan yang maksimal dari sejumlah banyaknya barang yang dibawa, dengan syarat berat muatan *container* tidak melebihi kapasitas *container* pengangkut, dan masing-masing *container* tersebut memiliki nilai atau *profit* yang tinggi. Inti dari permasalahan ini adalah bagaimana menentukan kombinasi barang yang akan diangkut untuk memperoleh keuntungan yang maksimal, dengan pertimbangan berat muatan tidak melebihi kapasitas alat angkut yang digunakan.

Persoalan optimasi pada pemilihan benda yang dapat dimasukkan ke dalam sebuah wadah yang memiliki keterbatasan ruang atau daya tampung disebut *knapsack*

problem. *Knapsack* atau ransel dimisalkan sebagai suatu wadah (tempat) dengan kapasitas M yang dapat memuat beberapa benda atau objek. Beberapa objek yang akan ditempatkan ke dalam wadah tersebut mempunyai berat dan volume yang berbeda-beda (Purwanto, 2008). Masalah *knapsack* merupakan sebuah persoalan yang sering dihadapi terutama pada bidang jasa pengangkutan barang seperti pengangkutan barang ke dalam peti kemas. Dalam usaha tersebut, diinginkan suatu keuntungan yang maksimal untuk mengangkut barang yang ada dengan tidak melebihi batas kapasitas yang ada. Berdasarkan persoalan tersebut, diharapkan ada suatu solusi yang secara otomatis dalam mengatasi persoalan itu. *Knapsack problem* adalah permasalahan optimasi kombinatorial, dimana kita harus mencari solusi terbaik dari banyak kemungkinan yang dihasilkan.

Dalam menyelesaikan masalah optimasi wadah atau dalam hal ini *knapsack problem* dapat digunakan metode heuristik, yaitu suatu metode pencarian yang didasarkan atas intuisi atau aturan-aturan empiris untuk memperoleh solusi yang lebih baik daripada solusi yang telah dicapai sebelumnya (Taha, 2002). Metode heuristik yang dapat diterapkan pada masalah optimasi misalnya adalah Metode *Hillclimbing*, Algoritma *Simulated Annealing*, dan Algoritma Genetika (Nallamottu, dkk, 2002).

Berdasarkan uraian di atas, dibutuhkan suatu aplikasi yang dapat membantu petugas di lapangan dalam optimasi seleksi barang sehingga dihasilkan *profit* yang maksimal. Selain itu, di era teknologi yang semakin maju dan berkembang pesat, dibutuhkan juga kinerja yang cepat, tepat dan efisien. Sehingga pemanfaatan teknologi

yang sudah dikembangkan, diharapkan dapat meningkatkan kinerja dan kualitas pelayanan penyedia jasa.

Di dalam Al-Qur'an pun telah membahas bagaimana Allah swt. menghendaki manusia memperoleh kemudahan setelah sebelumnya diuji oleh kesusahan. Hal ini dinyatakan oleh Allah swt. dalam firman-Nya dalam QS. Al-Insyirah/94:5-6 yang berbunyi:

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا ۚ إِنَّ مَعَ الْعُسْرِ يُسْرًا ۚ

Terjemahnya :

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan. Sesungguhnya sesudah kesulitan itu ada kemudahan”. (Departemen Agama RI, 2000).

Pada ayat di atas dijelaskan bahwa sesungguhnya tidak ada kesulitan yang tidak teratasi, jika jiwa seseorang bersemangat untuk keluar dari kesulitan dan mencari jalan pemecahan menggunakan akal pikiran yang benar dengan bertawakal pada Allah swt. Allah swt. dalam ayat ini bermaksud menjelaskan dalam salah satu sunnah-Nya yang bersifat umum dan konsisten, yaitu “setiap kesulitan pasti disertai atau disusul oleh kemudahan selama yang bersangkutan bertekad untuk menanggulangnya.” (Shihab, 2002).

Pada penyelesaian masalah *knapsack* dalam penelitian ini akan diterapkan algoritma genetika dan algoritma *greedy* untuk mencari solusi terbaik dari banyak kemungkinan yang dihasilkan. Dalam hukum genetika hanya individu yang berkualitas yang mampu menghasilkan individu atau generasi baru dengan kualitas terbaik. Hal ini sesuai dengan firman Allah swt. dalam Q.S Al-An'am/ 6: 6 yang berbunyi:

أَلَمْ يَرَوْا كَمْ أَهْلَكْنَا مِنْ قَبْلِهِمْ مِنْ قَرْنٍ مَكَّيْنَهُمْ فِي الْأَرْضِ مَا لَمْ نُمْكِنْ لَكُمْ
وَأَرْسَلْنَا السَّمَاءَ عَلَيْهِمْ مِدْرَارًا وَجَعَلْنَا الْأَنْهَارَ تَجْرِي مِنْ تَحْتِهِمْ فَأَهْلَكْنَاهُمْ
بِذُنُوبِهِمْ وَأَنْشَأْنَا مِنْ بَعْدِهِمْ قَرْنًا آخَرِينَ ٦

Terjemahnya:

“Apakah mereka tidak memperhatikan berapa banyaknya generasi-generasi yang telah Kami binasakan sebelum mereka, padahal (generasi itu), telah Kami teguhkan kedudukan mereka di muka bumi, yaitu keteguhan yang belum pernah Kami berikan kepadamu, dan Kami curahkan hujan yang lebat atas mereka dan Kami jadikan sungai-sungai mengalir di bawah mereka, kemudian Kami binasakan mereka karena dosa mereka sendiri, dan kami ciptakan sesudah mereka generasi yang lain”. (Departemen Agama RI, 2000).

Dalam tafsir Shihab (2002) dijelaskan bahwa: Dalam ayat ini Allah swt. memberi peringatan yang mengandung ajakan berfikir, siapa tahu mereka kembali dapat menempu jalan yang benar. Allah swt. memang terus-menerus memberikan peluang kepada manusia agar melakukan introspeksi sampai pada batas terakhir dari masa hidup manusia, atau sampai pada batas dimana yang bersangkutan benar-benar terbukti menolak ajakan Ilahi. Melalui ayat ini Allah swt. mengajak, apakah mereka tidak memperhatikan, yakni mempelajari sejarah atau mencari tahu, berapa banyak generasi yang telah kami binasakan sebelum mereka, padahal telah Kami teguhkan mereka, yakni generasi itu, di muka bumi yaitu dengan kekuatan jasmani, kelapangan, dan lain-lain, keteguhan yang belum pernah kami berikan kepada kamu, wahai masyarakat Makkah.

Dalam algoritma genetika pada generasi juga dilakukan mutasi, persilangan untuk menghasilkan generasi terbaik yang bernilai optimal. Sama dengan ayat di atas,

Allah swt. juga beberapa kali menguji generasi sebelumnya dengan memberikan nikmat lalu diuji dan kemudian dibinasakan untuk mengetahui apakah generasi tersebut benar-benar menolak ajaran Ilahi.

Dalam algoritma *greedy*, *knapsack problem* diselesaikan dengan memecahkan masalah langkah demi langkah dan merupakan salah satu metode dalam masalah optimasi. Prinsip dari algoritma *greedy* adalah “*take what you can get now*” yaitu mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan.

Penyelesaian *knapsack problem* pada penelitian ini, dibuat dengan menggunakan algoritma genetika dan algoritma *greedy*, hasil pencarian solusi dari keduanya akan dibandingkan untuk mendapatkan hasil yang lebih optimal. Diharapkan aplikasi ini dapat membantu dalam proses seleksi barang sehingga diperoleh keuntungan atau *profit* yang maksimal.

B. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan di atas, maka pokok permasalahan yang dihadapi adalah “Bagaimana menerapkan algoritma genetika dan perbandingannya dengan algoritma *greedy* dalam menyelesaikan *knapsack problem*?”

C. Fokus Penelitian dan Deskripsi Fokus

Agar dalam pengerjaan tugas akhir ini dapat lebih terarah, maka fokus penelitian penulisan ini difokuskan pada pembahasan sebagai berikut:

1. Aplikasi yang dirancang adalah aplikasi untuk menyelesaikan *knapsack problem*, dalam hal ini menentukan kombinasi barang yang akan dimasukkan

ke dalam peti kemas dengan menggunakan algoritma genetika dan algoritma *greedy*.

2. Aplikasi ini menyelesaikan masalah *unbounded knapsack problem*.
3. Aplikasi yang dirancang merupakan aplikasi berbasis web.
4. Target pengguna aplikasi ini adalah pengelola jasa peti kemas.
5. Aplikasi ini akan menampilkan solusi optimal dari hasil pencarian yang berupa terpilih atau tidaknya suatu barang untuk bisa dimasukkan ke dalam peti kemas dalam hal ini pemilihan barang LCL.
6. Berat atau volume total dari item barang yang dipilih tidak boleh melebihi kapasitas dari peti kemas.
7. Aplikasi ini menggunakan satuan kg dalam menghitung berat barang.
8. Aplikasi ini hanya menggunakan parameter berat dan profit dalam pemilihan barang sehingga tidak dapat memperhitungkan *space* kosong.

Sedangkan untuk mempermudah pemahaman dan memberikan gambaran serta menyamakan persepsi antara penulis dan pembaca, maka dikemukakan penjelasan yang sesuai dengan deskripsi fokus dalam penelitian ini. Adapun deskripsi fokus dalam penelitian ini adalah perancangan aplikasi dengan menerapkan atau menggunakan suatu cara atau metode dari algoritma genetika dan *greedy by profit* untuk menyelesaikan masalah *knapsack* dimana dalam hal ini menentukan kombinasi barang yang akan dimasukkan ke dalam peti kemas. Algoritma genetika adalah algoritma komputer yang mencari suatu solusi terbaik dalam permasalahan yang memiliki sejumlah besar kemungkinan pemecahan yang ada. Sedangkan algoritma

greedy yang akan digunakan adalah *greedy by profit* dimana strategi ini mengharapkan keuntungan sebesar-besarnya dengan memasukkan barang yang memiliki keuntungan per unit terbesar terlebih dahulu ke dalam wadah.

Knapsack problem adalah suatu masalah bagaimana menentukan pemilihan barang dari sekumpulan barang dimana setiap barang tersebut mempunyai berat dan *profit* masing-masing, sehingga dari pemilihan barang tersebut didapatkan *profit* yang maksimum. Dalam aplikasi ini akan diselesaikan *unbounded knapsack problem* dimana tidak ada batasan pengambilan jumlah sebuah item tetapi tetap memperhatikan batasan beban (*weight*).

Pada aplikasi yang akan dibuat, web adalah fokus yang penting dalam perancangannya. Web adalah salah satu layanan komunikasi interaktif yang berisi sekumpulan halaman yang menyediakan berbagai macam informasi yang dapat diakses oleh seluruh orang didunia selama terhubung dengan *internet*. Dengan web, sistem ini dapat diakses oleh *user* target. Adapun target pengguna aplikasi ini adalah pengelola jasa peti kemas.

Aplikasi ini akan menampilkan solusi optimal dari hasil pencarian yang berupa terpilih atau tidaknya suatu barang untuk bisa dimasukkan ke dalam peti kemas. Adapun hasil dari penerapan algoritma genetika dan *greedy* dalam aplikasi adalah dapat membantu petugas di lapangan dalam optimasi seleksi barang LCL sehingga dihasilkan *profit* yang maksimal. LCL adalah singkatan dari *Less Than Container Load* dimana pengiriman barang dapat dilakukan dalam jumlah sedikit atau tidak

sampai penuh satu *container*. Adapun barang yang dikirim bisa berasal dari satu pengirim atau dari beberapa pengirim dimana barang yang dikirim adalah campuran dari berbagai macam komoditi.

Persoalan optimasi pada pemilihan benda yang dapat dimasukkan ke dalam sebuah wadah yang memiliki keterbatasan ruang atau daya tampung disebut *knapsack problem* (Purwanto, 2008). Adapun syarat dari solusi *knapsack problem* ini adalah berat atau volume total dari item barang yang dipilih tidak boleh melebihi kapasitas dari wadah yang dalam hal ini adalah peti kemas.

Dalam aplikasi ini digunakan satuan kg dalam menghitung berat barang. Sedangkan dalam pemilihan kombinasi barang yang akan dimasukkan ke dalam peti kemas digunakan parameter berat dan *profit*. Sehingga aplikasi ini tidak dapat memperhitungkan kondisi barang dalam peti kemas, tetapi hanya dapat melakukan pemilihan barang berdasarkan berat dan *profitnya*.

D. Kajian Pustaka/ Penelitian Terdahulu

Perkembangan teknologi informasi yang semakin maju banyak terdapat aplikasi yang berhubungan dengan *knapsack problem*. Akan tetapi metode yang digunakan berbeda-beda serta penggunaan teknologi yang beraneka ragam. Beberapa aplikasi yang pernah dibuat antara lain :

Penelitian pertama oleh Maria Irmina Prasetyowati dan Arya Wicaksana (2013), yang berjudul “*Implementasi Algoritma Dynamic Programming untuk Multiple Constraints Knapsack Problem*”. Penelitian ini bertujuan untuk menghasilkan aplikasi

optimasi untuk melakukan pencarian solusi optimal dari *Multiple Constraints Knapsack Problem* dengan studi kasus pemilihan media promosi di UMN dan untuk mengimplementasikan algoritma *Dynamic Programming* pada aplikasi sehingga memudahkan pengguna dalam pemilihan media promosi di UMN yang akan digunakan.

Penelitian tersebut memiliki kesamaan dengan aplikasi yang akan dibuat yaitu sama-sama mengangkat masalah *knapsack problem* tetapi baik studi kasus dan metode penyelesaiannya berbeda. Pada penelitian tersebut menggunakan algoritma *Dynamic Programming* untuk mencari solusi optimal dari *Multiple Constraints Knapsack Problem* dengan studi kasus pemilihan media promosi di UMN, sedangkan penelitian ini menggunakan algoritma genetika dan *greedy* untuk menyelesaikan *unbounded knapsack problem* dengan studi kasus optimasi pengangkutan barang pada peti kemas.

Penelitian kedua oleh Ari Agustina, Hanifa Vidya Rizanti, Arianty Anggraini, dan Winda Ayu Irianto (2012), dengan judul “*Perbandingan Algoritma Exhaustive Search dan Algoritma Genetika untuk Memecahkan Knapsack Problem*”. Penelitian ini akan menganalisa pemakaian algoritma *Exhaustive Search* atau sering disebut *Brute Force* pada kasus *knapsack problem* dalam kaitannya dengan permasalahan *knapsack* dan melakukan perbandingan dengan algoritma genetika.

Penelitian ini memiliki kesamaan dengan penelitian di atas yaitu sama-sama menggunakan algoritma genetika dalam penyelesaian *knapsack problem*. Namun penelitian tersebut lebih kepada melakukan analisa dan perbandingan penggunaan algoritma *Exhaustive Search* dengan algoritma genetika dalam kaitannya dengan

permasalahan *knapsack*, sedangkan penelitian ini menggunakan algoritma genetika untuk menyelesaikan *unbounded knapsack problem* dan menampilkan solusi optimal dari hasil pencarian yang berupa terpilih atau tidaknya suatu barang untuk bisa dimasukkan ke dalam peti kemas.

Penelitian ketiga oleh Daniel Jahja Surjawan dan Irene Susanto (2009), dengan judul “*Aplikasi Optimalisasi Muat Barang dengan Penerapan Algoritma Dynamic Programming pada Persoalan Integer Knapsack*”. Penelitian ini menitikberatkan pada optimalisasi dan *tracking* pada pengiriman barang secara *desktop*, dan pencegahan *overload* saat memuat barang dengan menerapkan algoritma *Dynamic Programming*.

Jika penelitian di atas menitikberatkan pada penggunaan algoritma *Dynamic Programming* dalam optimalisasi dan *tracking* dalam pengiriman barang dan mencegah terjadinya *overload* saat memuat barang, maka penelitian ini menggunakan algoritma genetika untuk menyeleksi barang-barang yang akan dimasukkan ke dalam peti kemas sehingga didapat generasi terbaik yang optimum dari sejumlah generasi tersebut. Kombinasi barang yang terbaik inilah yang dimasukkan ke dalam peti kemas sehingga tidak akan melebihi kapasitas yang ada.

E. Tujuan dan Kegunaan Penelitian

1. Tujuan penelitian

Penelitian ini bertujuan untuk merancang dan membuat aplikasi optimasi bagaimana menentukan kombinasi barang ke dalam peti kemas atau *container* dengan menggunakan algoritma genetika dan *greedy* dalam menyeleksi barang ke dalam peti kemas, sehingga diperoleh keuntungan atau *profit* yang maksimal.

2. Kegunaan Penelitian

a. Kegunaan bagi dunia akademik

Memberikan referensi yang berguna bagi dunia akademis khususnya dalam penelitian yang akan dilaksanakan oleh para peneliti yang akan datang dalam hal perkembangan teknologi berbasis web dan penerapan algoritma genetika, algoritma *greedy* dan penyelesaian *knapsack problem*.

b. Kegunaan bagi pengguna

Memudahkan dalam melakukan seleksi barang untuk menentukan kombinasi barang yang akan diangkut ke peti kemas sehingga memperoleh keuntungan atau *profit* yang maksimal.

c. Kegunaan bagi penulis

Mengembangkan wawasan keilmuan dan meningkatkan pemahaman tentang struktur dan sistem kerja dalam pengembangan aplikasi berbasis web, penyelesaian *knapsack problem* dengan algoritma genetika dan *greedy*, menambah pengetahuan penulis mengenai pengelolaan jasa peti kemas dan mengajarkan penulis bahwa akan ada kemudahan dari Allah bagi orang-orang yang yakin dan ingin berusaha. Serta menjadi salah satu syarat untuk menyelesaikan studi strata satu jurusan teknik informatika UIN Alauddin Makassar.

BAB II

TINJAUAN TEORITIS

A. Tinjauan Al-Qur'an dan Hadist

Manusia adalah makhluk ciptaan Allah swt. yang sempurna karena dibekali akal dan pikiran yang tidak dimiliki oleh makhluk ciptaan lainnya. Akal dan pikiran memegang peranan penting dalam kehidupan manusia. Dengan kemampuan berpikir, manusia dapat mempelajari berbagai ilmu yang dapat dimanfaatkannya dalam kehidupan sehari-hari.

Adapun Al-Hadist yang mewajibkan manusia untuk menuntut ilmu adalah :

طَلَبُ الْعِلْمِ فَرِيضَةٌ عَلَى كُلِّ مُسْلِمٍ وَمُسْلِمَةٍ

Artinya :

“Mencari ilmu itu adalah wajib bagi setiap muslim laki-laki maupun muslim perempuan”. (HR. Ibnu Abdil Barr)

Dengan adanya ilmu pengetahuan yang dimiliki oleh manusia maka teknologi akan semakin maju dan dapat meringankan pekerjaan manusia, selama teknologi itu digunakan dengan benar. Selain itu dengan adanya ilmu pengetahuan manusia akan lebih mudah terhindar dari adanya dampak-dampak negatif teknologi.

Seperti dalam usaha pengelolaan peti kemas, seiring bertambahnya ilmu pengetahuan maka dapat pula digunakan teknologi dalam pengelolaannya sehingga dapat memudahkan dan memaksimalkan keuntungannya. Tak dapat dipungkiri saat ini dibutuhkan juga kinerja yang cepat, tepat dan efisien, sehingga pemanfaatan teknologi

yang sudah dikembangkan, diharapkan dapat meningkatkan kinerja dan kualitas pelayanan penyedia jasa.

Hal ini sesuai dengan firman Allah swt. dalam surah Al-Baqarah ayat 185 :

يُرِيدُ اللَّهُ بِكُمُ الْيُسْرَ وَلَا يُرِيدُ بِكُمُ الْعُسْرَ

Terjemahnya :

“...Allah menghendaki kemudahan bagimu, dan tidak menghendaki kesukaran bagimu....” (Departemen Agama RI, 2000).

Dalam tafsir Shihab (2002) dijelaskan bahwa: Ayat ini menetapkan siapa yang wajib berpuasa, barang siapa diantara kamu hadir pada bulan itu, yakni berada di negeri tempat tinggalnya atau mengetahui munculnya awal bulan Ramadhan sedang dia tidak berhalangan dengan halangan yang dibenarkan agama, maka hendaklah ia berpuasa pada bulan itu. Dalam ayat ini juga berisi penjelasan barang siapa yang sakit atau dalam perjalanan lalu ia berbuka maka wajiblah baginya berpuasa sebanyak hari yang ditinggalkannya itu, pada hari-hari yang lain. Bolehnya musafir dan orang sakit untuk tidak berpuasa adalah bukti bahwa Allah memberikan kemudahan kepada hamba-Nya dalam syariat-Nya.

Sebagaimana Nabi Muhammad saw bersabda :

يَسِّرَا وَلَا تُعَسِّرَا، وَبَشِّرَا وَلَا تُنْفِرَا، وَتَطَاوَعَا وَلَا تَخْتَلِفَا

Artinya :

“Hendaknya kalian mempermudah dan jangan mempersulit, berikanlah kabar gembira dan jangan membuat lari, saling membantu dan jangan berselisih” (HR. Bukhari 3038 dan Muslim 1733).

Islam sangat memperhatikan keperluan manusia dan memelihara kebahagiaan yang dimilikinya. Oleh sebab itu, hukum Islam seluruhnya senantiasa berada dalam batas-batas kemampuan dan kekuatan manusia. Tidak ada satu hukum pun yang tidak mampu dilaksanakan oleh manusia. Dan apabila seorang muslim menemui suatu kesulitan di luar kemampuannya, atau kesukaran yang disebabkan oleh suatu sebab tertentu, maka Islam membuka di hadapannya pintu keringanan dan memberi kemurahan.

B. Aplikasi

Aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolahan kata, lembar kerja, dan pemutar media (Wikipedia, 2015).

Pengertian Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah

satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan maupun pemrosesan data yang diharapkan. Secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya, aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user (Hasugian, 2012).

C. Optimasi

Optimasi adalah suatu proses untuk mencapai hasil yang ideal atau optimasi (nilai efektif yang dapat dicapai). Optimasi dapat diartikan sebagai suatu bentuk mengoptimalkan sesuatu hal yang sudah ada, ataupun merancang dan membuat sesuatu secara optimal. (Wikipedia, 2016)

Fokus dari optimasi adalah menemukan solusi yang optimal pada masalah desain melalui pertimbangan sistematis dari alternatif yang diberikan untuk kepuasan sumber daya dan kendala biaya. Tujuan optimasi secara keseluruhan untuk meminimalkan biaya, memaksimalkan keuntungan, untuk merampingkan kegiatan produksi, meningkatkan efisiensi proses, dan lain-lain. Dalam menentukan solusi yang optimal diperlukan pertimbangan yang cermat dari beberapa alternatif yang sering dibandingkan pada beberapa kriteria. Permasalahan optimasi adalah suatu permasalahan yang mempunyai banyak solusi dan harus bisa ditentukan solusi mana yang dikatakan optimal.

D. Peti Kemas

Peti kemas adalah peti besar yang dipersiapkan untuk diisi barang yang akan dikirim supaya tidak rusak. (KBBI, 2016)

Peti kemas (*container*) adalah satu kemasan yang dirancang secara khusus dengan ukuran tertentu, dapat dipakai berulang kali, dipergunakan untuk menyimpan dan sekaligus mengangkut muatan yang ada di dalamnya. Filosofi di balik petikemas ini adalah adanya kemasan yang terstandar yang dapat dipindah-pindahkan ke berbagai moda transportasi laut dan darat dengan mudah seperti kapal laut, kereta api, truk atau angkutan lainnya sehingga transportasi ini efisien, cepat, aman dan kalau mungkin diangkut dari pintu ke pintu (*door to door*).

Ukuran peti kemas (*container*) yang distandarisasikan oleh *International Standard Organization (ISO)* ada dua *container* yaitu ukuran 200 *feet* dengan kapasitas 22.1 ton dan ukuran 40 *feet* dengan kapasitas 27,396 ton. Sementara *container 40 high cube dry* memiliki kapasitas 29,6 ton. Adapun jenis peti kemas dibedakan menurut jenis muatannya.

E. Knapsack Problem

Persoalan optimasi pada pemilihan benda yang dapat dimasukkan ke dalam sebuah wadah yang memiliki keterbatasan ruang atau daya tampung disebut *knapsack problem*. *Knapsack* atau ransel dimisalkan sebagai suatu wadah (tempat) dengan kapasitas M yang dapat memuat beberapa benda atau objek. Beberapa objek yang akan ditempatkan ke dalam wadah tersebut mempunyai berat dan volume yang berbeda-beda (Purwanto, 2008).

Knapsack problem merupakan masalah optimasi kombinatorial. Sebagai contoh adalah suatu kumpulan barang masing-masing memiliki berat dan nilai, kemudian akan ditentukan jumlah tiap barang untuk dimasukkan dalam koleksi sehingga total berat

kurang dari batas yang diberikan dan nilai total seluas mungkin (Wikipedia, 2009). Dapat dikatakan bahwa *knapsack problem* merupakan masalah di mana orang dihadapkan pada persoalan optimasi pada pemilihan benda yang dapat dimasukkan ke dalam sebuah wadah yang memiliki keterbatasan ruang atau daya tampung. Dengan adanya optimasi dalam pemilihan benda yang akan dimasukkan ke dalam wadah tersebut diharapkan dapat menghasilkan keuntungan yang maksimum.

F. Algoritma Genetika

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Algoritma genetika pertama kali dikembangkan oleh John Holland dari Universitas Michigan (1975). John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami maupun buatan) dapat diformulasikan dalam terminologi genetika. Algoritma genetika adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom (Kusumadewi, 2003). Algoritma genetika merupakan algoritma pencarian yang didasarkan atas mekanisme dari seleksi alam yang lebih dikenal dengan proses evolusi. Dalam proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya. Hanya individu-individu yang kuat yang mampu bertahan. Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembangbiakan. Dalam algoritma genetika, proses perkembangbiakan ini menjadi proses dasar yang menjadi perhatian utama, dengan dasar berfikir: “Bagaimana mendapatkan keturunan yang lebih baik” (Basuki, 2003).

Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin, yang dikenal dengan istilah *populasi*. Individu yang terdapat dalam satu populasi disebut dengan istilah *kromosom*. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan istilah *generasi*. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) terbentuk dari dua gabungan dua kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Selain operator penyilangan, suatu kromosom dapat juga dimodifikasi dengan menggunakan operator *mutasi*. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai *fitness* dari kromosom induk (*parent*) dan nilai *fitness* dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik (Kusumadewi,2003).

Algoritma genetika merupakan salah satu algoritma yang sangat tepat digunakan untuk penyelesaian masalah optimasi yang kompleks dan sukar diselesaikan dengan menggunakan metode yang konvensional. Algoritma genetik merupakan teknik pencarian yang di dalam ilmu komputer untuk menemukan penyelesaian perkiraan untuk optimisasi dan masalah pencarian. Sehingga algoritma genetika ini banyak

dipakai pada aplikasi bisnis, teknik maupun pada bidang keilmuan. Algoritma ini dapat dipakai untuk mendapatkan solusi yang tepat untuk masalah optimal dari satu variabel atau multi variabel. Suatu algoritma genetika yang sederhana umumnya terdiri dari tiga operator yaitu: operator reproduksi, operator *crossover* (persilangan) dan operator mutasi. Secara sederhana, algoritma umum dari algoritma genetik ini dapat dirumuskan menjadi beberapa langkah, yaitu:

1. Membentuk suatu populasi individual dengan keadaan acak
2. Mengevaluasi kecocokan setiap individual keadaan dengan hasil yang diinginkan
3. Memilih individual dengan kecocokan yang tertinggi
4. Bereproduksi, mengadakan persilangan antar individual terpilih diselingi mutasi
5. Mengulangi langkah 2-4 sampai ditemukan individual dengan hasil yang diinginkan.

G. Algoritma Greedy

Algoritma *Greedy* adalah algoritma yang memecahkan masalah langkah demi langkah dan merupakan salah satu metode dalam masalah optimasi. Prinsip dari algoritma *greedy* adalah “*take what you can get now*” yaitu mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan. Algoritma *greedy* membentuk solusi langkah per langkah sebagai berikut:

1. Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi.
Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik

dalam menentukan pilihan. Keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah selanjutnya.

2. Pendekatan yang digunakan di dalam algoritma *greedy* adalah membuat pilihan yang terlihat memberikan perolehan terbaik, yaitu dengan membuat pilihan optimum lokal pada setiap langkah dan diharapkan akan mendapatkan solusi optimum global.

Algoritma *greedy* didasarkan pada pemindahan *edge* per *edge* dan pada setiap langkah yang diambil tidak memikirkan konsekuensi ke depan, *greedy* tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada serta sebagian masalah *greedy* tidak selalu berhasil memberikan solusi yang benar-benar optimum tapi pasti memberikan solusi yang mendekati nilai optimum. Pada *greedy algorithm* ada beberapa strategi yang digunakan untuk memilih objek yang akan dimasukkan ke dalam *knapsack* yaitu *greedy by profit*, *greedy by weight*, dan *greedy by density*. *Greedy by profit* mengharapkan keuntungan maksimal dengan cara memasukkan barang atau objek dengan nilai keuntungan terbesar terlebih dahulu ke dalam kantong atau *knapsack*. Jadi strategi ini hanya mempertimbangkan jumlah keuntungan dari sekumpulan barang, dengan catatan berat barang yang akan dibawa tidak melebihi kapasitas kantong. *Greedy by weight*, memasukkan barang sebanyak-banyaknya ke dalam kantong, jadi barang atau objek yang dimasukkan terlebih dahulu adalah barang dengan bobot paling ringan terlebih dahulu, dengan harapan dengan banyaknya barang atau objek yang terangkut bis mendapatkan keuntungan yang sebesar-besarnya. *Greedy by density*, strategi ini mengharapkan keuntungan sebesar-besarnya dengan

memasukkan barang yang memiliki keuntungan per unit terbesar (P_i/W_i) terlebih dahulu ke dalam kantong atau *knapsack*.

H. Web

Web atau *World Wide Web* adalah salah satu fasilitas yang disediakan diinternet. Web merupakan dunia maya di internet yang terdapat ribuan informasi tentang segala hal dan dikemas dalam bentuk dokumen *hypertext* (Nugroho, 2007).

Web adalah salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet. Web ini menyediakan informasi bagi pemakai komputer yang terhubung ke internet dari sekedar informasi “sampah” atau informasi yang tidak berguna sama sekali sampai informasi yang serius, dari informasi yang gratisan sampai informasi yang komersial. *Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Cara kerja web adalah informasi web disimpan dalam dokumen dalam bentuk halaman-halaman web atau *web page*. Halaman web tersebut disimpan dalam komputer *server* web sementara dipihak pemakai ada komputer yang bertindak sebagai komputer *client* dimana ditempatkan program untuk membaca halaman web yang ada di *server web* (*browser*), kemudian *browser* membaca halaman web yang ada di *server web*.

I. Hypertext Markup Language (HTML)

HTML adalah salah satu format yang digunakan untuk menulishalaman web, HTML ini berjalan di web *browser* dan memiliki fungsi untuk melakukan pemrograman aplikasi di atas web. HTML merupakan pengembangan dari standar pemformatan dokumen teks, yaitu standar *Generalized Markup Language*. HTML sebenarnya adalah dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu (Kadir, 2002).

J. Hypertext Preprocessor (PHP)

PHP adalah *script* untuk pemrograman *script web server-side*, *script* yang membuat dokumen HTML secara *on the fly*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. Dengan menggunakan PHP maka *maintenance* suatu situs web menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP.

PHP merupakan singkatan *rekursif* (akronim berulang) dari PHP *Hypertext Preprocessor*. PHP adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini atau dalam kata lain bisa diartikan sebuah bahasa pemrograman *web* yang bekerja di sisi *server* (*server side scripting*) yang dapat melakukan konektifitas pada database yang dimana hal itu tidak dapat dilakukan hanya dengan menggunakan sintaks-sintaks HTML biasa. PHP banyak dipakai untuk memprogram situs *web* dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. (Rosari, 2008).

K. MySQL

MySQL adalah *multiuser database* yang menggunakan bahasa *Structured Query Language* (SQL). *MySQL* dalam operasi *client-server* melibatkan *server daemon MySQL* di sisi *server* dan berbagai macam program serta *library* yang berjalan di sisi *client*. *MySQL* mampu menangani data yang cukup besar. Perusahaan yang mengembangkan *MySQL* yaitu TcX, mengaku mampu menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 *Gigabyte* data. (Sunarfri hantono, 2002).

L. Cascading Style Sheet (CSS)

Cascading Style Sheet (CSS) merupakan suatu bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur dan seragam. CSS saat ini dikembangkan oleh *World Wide Web Consortium* (W3C) dan menjadi bahasa standar dalam pembuatan web. CSS difungsikan sebagai penopang atau pendukung, dan pelengkap dari file html yang berperan dalam penataan kerangka dan layout.

Dengan menggunakan CSS, akan banyak keuntungan yang dapat diperoleh. Diantaranya adalah memisahkan pembuatan dokumen (CSS dan HTML), mempermudah pembuatan dan pemeliharaan dokumen web, mempercepat pembacaan HTML, fleksibel, interaktif, tampilan lebih menarik dan dapat digunakan pada semua web *browser*. (Agus dan Feni Agustin, 2011).

M. Javascript

Javascript adalah bahasa yang berfungsi untuk membuat skrip-skrip program yang dapat dikenal dan dieksekusi oleh *web browser* dengan tujuan untuk menjadikan halaman web lebih bersifat interaktif. Meskipun banyak fitur dari bahasa java yang diadopsi oleh *javascript*, namun *javascript* dikembangkan secara terpisah dan independen. *Javascript* dikembangkan oleh *Netscape* dan merupakan bahasa yang bersifat terbuka sehingga setiap orang dapat menggunakannya tanpa harus membeli lisensi.

Beberapa contoh penggunaan *javascript* yang sering dijumpai dalam halaman web adalah untuk menampilkan pesan peringatan, untuk menampilkan *popup window*, untuk membuat menu *dropdown*, untuk melakukan validasi pada saat *user* memasukkan data dalam suatu *form* dan untuk menampilkan tanggal dan waktu. (Raharjo, 2011).


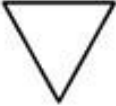

N. Daftar Simbol

1. Flowmap

Flowmap atau bagan alir adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program atau prosedur sistem secara logika. *Flowmap* ini berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *flowmap* ini harus dapat memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi. Terdapat simbol-simbol standar dalam *flowmap* yaitu terlihat pada tabel II.1.

Tabel II. 1. Daftar Simbol *Flowmap* Diagram (Sakinah, 2002)





Simbol	Nama	Keterangan
	Terminator awal / akhir program	Untuk memulai dan mengakhiri suatu program
	Dokumen	Menunjukkan dokumen berupa dokumen input dan output pada proses manual dan proses berbasis computer
	Proses Manual	Menunjukkan proses yang dilakukan secara manual.
	Proses Komputer	Menunjukkan proses yang dilakukan secara komputerisasi
	Arah aliran data	Menunjukkan arah aliran dokumen antar bagian yang terkait pada suatu sistem.
	Penyimpanan Manual	Menunjukkan media penyimpanan data / infomasi secara manual
	Data	Simbol input/output digunakan untuk mewakili data input/output



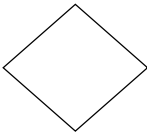

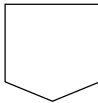
	File Harddisk	Media penyimpanan dari proses entry data dan proses komputerisasi
	Offline Stronge	Media penyimpanan data berupa arsip
	Keyboard	Proses penyimpanan menggunakan keyboard

2. Flowchart

Flowchart atau *Bagan alir* adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.

Tabel II.2 Daftar Simbol *Flowchart* (Kristanto, 2003)

Simbol	Nama	Keterangan
	<i>Terminator</i>	Permulaan atau akhir program
	<i>Flow Line</i>	Arah aliran program
	<i>Preparation</i>	Proses inisialisasi atau pemberian harga awal
	<i>Process</i>	Proses perhitungan atau proses pengolahan data

	<i>Input/Output Data</i>	Proses input atau output data, parameter, informasi
	<i>Predefined Process</i>	Permulaan sub program atau proses menjalankan sub program
	<i>Decision</i>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<i>On Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang ada pada satu halaman
	<i>Off Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> yang ada pada halaman berbeda

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Dalam melakukan penelitian ini, jenis penelitian kualitatif yang digunakan adalah *Design and Creation*. Dipilihnya jenis penelitian ini oleh penulis dikarenakan konsep dari *Design and Creation* sangat tepat untuk mengelola penelitian ini. Disamping melakukan penelitian tentang judul ini, penulis juga mengembangkan produk berdasarkan penelitian yang dilakukan.

B. Pendekatan Penelitian

Penelitian ini menggunakan pendekatan penelitian saintifik yaitu pendekatan berdasarkan ilmu pengetahuan dan teknologi.

C. Sumber Data

Sumber data pada penelitian ini adalah wawancara dengan narasumber yakni pengelola perusahaan jasa peti kemas. Selain itu data juga diperoleh dari buku pustaka terkait tentang *algoritma genetika*, *knapsack problem*, dan *algoritma greedy*, jurnal penelitian terdahulu yang memiliki keterkaitan pada penelitian ini dan sumber-sumber data *online* atau internet yang bisa dijadikan sumber informasi.

D. Metode Pengumpulan Data

Metode pengumpulan data yang dipakai pada penelitian untuk aplikasi ini adalah metode observasi dan studi literatur.

1. Observasi

Observasi yang dilakukan yaitu mengadakan kunjungan dan tanya jawab secara langsung kepada pihak yang mengelola jasa peti kemas pada Kawasan Logistik Terpadu Makassar, antara lain dengan :

- a. Apakah faktor- faktor penentu dalam menentukan terpilih atau tidaknya barang untuk diangkut?
- b. Apakah hambatan-hambatan dalam optimasi barang?
- c. Berapa lama waktu yang dibutuhkan untuk menentukan kombinasi barang yang akan diangkut?

2. Studi Literatur

Studi literatur adalah salah satu metode pengumpulan data dengan cara membaca buku-buku dan jurnal sesuai dengan data yang dibutuhkan. Pada penelitian ini, dipilih studi literatur untuk mengumpulkan referensi dari jurnal-jurnal yang memiliki kemiripan dalam pembuatan aplikasi ini.

E. Instrumen Penelitian

Adapun instrument penelitian yang digunakan dalam penelitian yaitu:

1. Perangkat Keras

Perangkat keras yang digunakan untuk mengembangkan dan mengumpulkan data pada aplikasi ini adalah sebagai berikut:

- a). Laptop Asus dengan spesifikasi :

- 1) Prosesor Intel Core i5

- 2) Harddisk 500 GB

3) *Memory* 4 GB

4) *VGA NVIDIA GEFORCE 720m*

2. Perangkat Lunak

Adapun perangkat lunak yang digunakan dalam aplikasi ini adalah sebagai berikut:

- 1) Sistem Operasi Windows 10
- 2) Notepad++
- 3) MySQL
- 4) Mozilla Firefox
- 5) XAMPP

F. Teknik Pengolahan dan Analisis Data

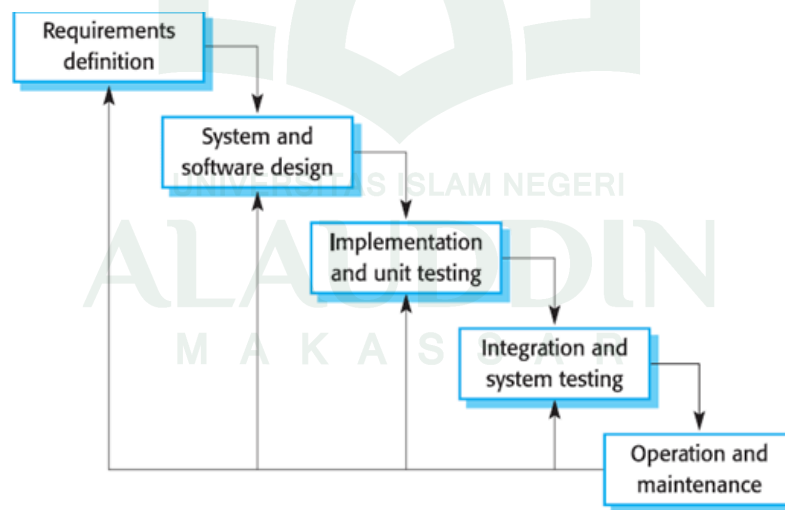
Analisis data terbagi menjadi dua yaitu, metode analisis kuantitatif dan metode analisis kualitatif. Analisis kuantitatif ini menggunakan data statistik dan dapat dilakukan dengan cepat, sementara analisis kualitatif ini digunakan untuk data kualitatif. Data yang digunakannya adalah berupa catatan-catatan yang biasanya cenderung banyak dan menumpuk sehingga membutuhkan waktu yang cukup lama untuk dapat menganalisisnya secara seksama.

Dalam penelitian ini, penulis menggunakan metode analisis kualitatif. Analisis kualitatif adalah prosedur penelitian yang bermaksud untuk memahami fenomena tentang apa yang dialami oleh subjek penelitian misalnya perilaku, persepsi, motivasi, tindakan, dan lain-lain. Secara holistik, dengan cara deskriptif dalam bentuk kata-kata dan bahasa (Moelong, 2002).

G. Metode Perancangan Aplikasi

Pada penelitian ini, metode perencanaan aplikasi yang digunakan adalah *Waterfall*. Model *Waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software, dimana proses pengerjaannya bertahap dan harus menunggu tahap sebelumnya selesai dilaksanakan kemudian memulai tahap selanjutnya.

Metode ini dipilih dikarenakan proses perancangan aplikasi dilakukan tahap demi tahap dimulai dari *Requirements analysis and definition*, *System and Software design*, *Implementation*, *Integration and System testing* dan *Operation and maintenance* (Pressman, 2010).



Gambar III.1 Model *Waterfall* (Pressman, 2010).

Berikut ini adalah deskripsi dari tahap model *Waterfall*:

1. *Requirements analysis and definition*

Proses menganalisis kebutuhan sistem kemudian pengumpulan kebutuhan secara lengkap yang sesuai dengan sistem yang akan dibangun, sehingga nantinya sistem yang telah dibangun dapat memenuhi semua kebutuhan.

2. *System and Software design*

Desain dikerjakan setelah analisis dan pengumpulan data dikumpulkan secara lengkap. Proses ini berfokus pada: struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural.

3. *Implementation*

Proses menerjemahkan desain ke dalam suatu bahasa yang bisa dimengerti oleh komputer.

4. *Integration and System testing*

Proses pengujian dilakukan pada logika, untuk memastikan semua pernyataan sudah diuji. Lalu dilanjutkan dengan melakukan pengujian fungsi sistem untuk menemukan kesalahan-kesalahan dan memastikan bahwa input akan memberikan hasil yang aktual sesuai yang dibutuhkan.

5. *Operation and maintenance*

Pada proses ini dilakukan pengoperasian sistem yang telah selesai dibangun dan melakukan pemeliharaan.

H. Teknik Pengujian

Proses pengujian perangkat lunak bertujuan untuk melihat sejauh mana perangkat lunak dapat berjalan. Adapun teknik pengujian sistem yang akan digunakan pada penelitian ini adalah *White Box Testing* dan perbandingan algoritma *greedy* dan genetika. *White box testing* adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara *procedural* untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan *white box testing* merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

Adapun teknik pengujian sistem yang digunakan yaitu *white box* dengan menggunakan metode *Cyclomatic Complexity* (CC). Dalam menguji suatu sistem, bagan alir program (*flowchart*) yang didesain sebelumnya dipetakan ke dalam bentuk bagan alir control (*flowgraph*). Hal ini memudahkan untuk penentuan jumlah region, *Cyclomatic Complexity* (CC) dan *independent path*. Jika jumlah region, *Cyclomatic Complexity* (CC) dan *independent path* sama besar maka sistem dinyatakan benar, tetapi jika sebaliknya maka sistem masih memiliki kesalahan, mungkin dari segi logika maupun dari sisi lainnya. *Cyclomatic Complexity* (CC) dapat dihitung dengan menggunakan rumus :

$$V(G) = E - N + 2$$

Dimana:

E = jumlah *edge* pada *flowgraph*

N = jumlah *node* pada *flowgraph*

Sedangkan perbandingan algoritma *greedy* dan genetika adalah pengujian perangkat lunak dengan menguji kedua algoritma dengan membandingkan beberapa hal yang dapat menunjukkan hasil penerapan algoritma genetika dalam menyelesaikan *knapsack problem* dan perbandingan algoritma genetika dan *greedy* dalam menyelesaikan masalah *knapsack* dengan optimal.

Rancangan Tabel Pengujian adalah sebagai berikut:

Tabel III. 1 Rancangan Tabel Pengujian White Box Testing

No.	Nama Modul/Menu	Hasil yang di harapkan		
		CC	Region	Path
1.	Pengujian Algoritma <i>Greedy</i>	Bernilai sama	Bernilai sama	Bernilai sama
2.	Pengujian Algoritma Genetika	Bernilai sama	Bernilai sama	Bernilai sama

Tabel III. 2 Rancangan Tabel Pengujian Perbandingan Algoritma

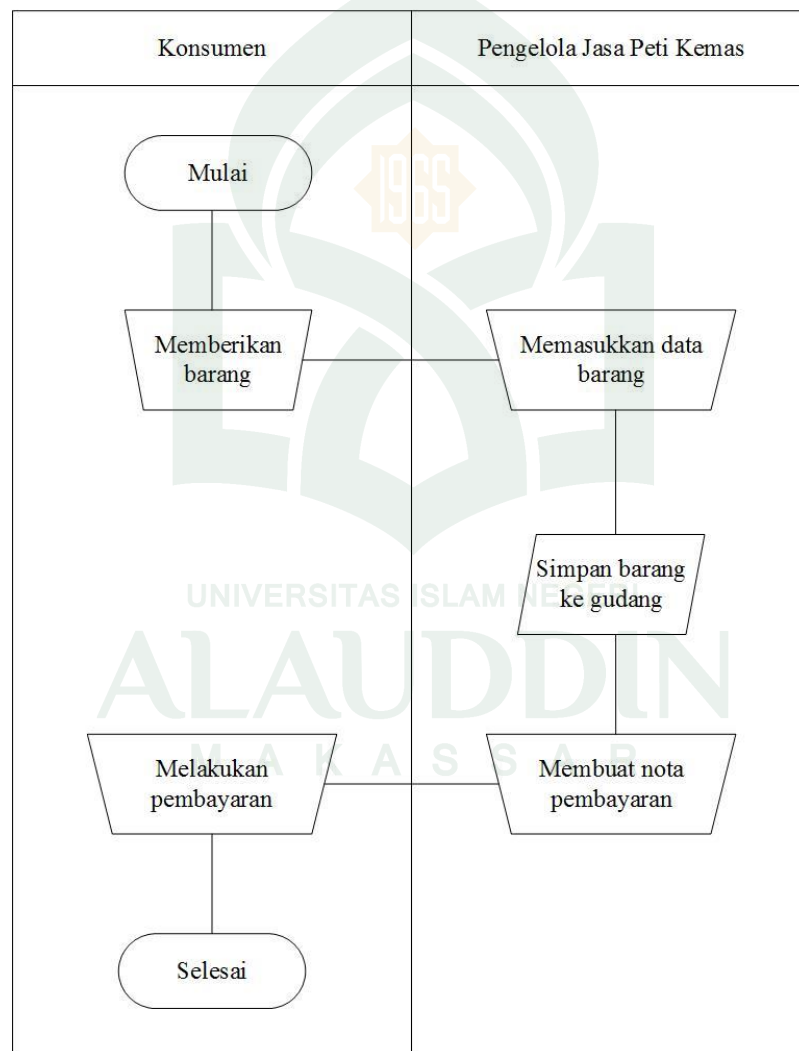
Algoritma	Indikator Pengujian		
	Solusi yang Dihasilkan	Maksimal Profit	Kecepatan Proses Optimasi
Greedy by Profit	Banyaknya jumlah barang yang terpilih	Besar profit yang dihasilkan	Waktu yang dibutuhkan untuk proses optimasi
Greedy by Weight	Banyaknya jumlah barang yang terpilih	Besar profit yang dihasilkan	Waktu yang dibutuhkan untuk proses optimasi
Genetika	Banyaknya jumlah barang yang terpilih	Besar profit yang dihasilkan	Waktu yang dibutuhkan untuk proses optimasi

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

A. Analisis Sistem yang Sedang Berjalan

Berikut ini adalah *flowmap* proses transaksi penggunaan jasa pengangkutan barang ke dalam peti kemas :



Gambar IV.1 *Flowmap* Analisis Sistem yang Sedang Berjalan

Konsumen memberikan sejumlah barang yang akan dikirim melalui jasa peti kemas. Kemudian barang tersebut diterima dan diproses oleh pihak pengelola jasa peti kemas dengan mencatat data dari barang seperti nama barang, berat dan tujuannya kemana. Kemudian barang tersebut disimpan di gudang sampai barang yang akan dikirim memenuhi kuota untuk pengiriman satu *container*. Pihak pengelola peti kemas memberikan dokumen berupa nota pembayaran kepada konsumen. Selanjutnya konsumen melakukan pembayaran sesuai dengan tarif.

B. Analisis Sistem yang diusulkan

1. Analisis Masalah

Dalam menjalankan proses bisnisnya, salah satu kendala yang dihadapi oleh perusahaan peti kemas adalah kurangnya kontrol dalam melakukan muat barang. Masalah terjadi saat melakukan muat barang sebanyak mungkin ke dalam *container*, asal angkut barang hingga terkadang melebihi kapasitas muatan. Selain itu dalam jasa pengangkutan barang atau dalam hal ini perusahaan peti kemas menginginkan keuntungan yang maksimal untuk setiap transaksi pengangkutan barang yang dilakukan. Namun, peti kemas (*container*) memiliki keterbatasan wadah dalam menampung barang sehingga bukan hanya sekedar pengangkutan saja yang harus dipikirkan, melainkan juga efisiensi dan keuntungan yang yang dapat diperoleh. Sehingga dibutuhkan keterampilan dalam penyusunan barang agar kapasitas ruang bisa optimal tanpa melebihi kemampuan daya tampung. Dengan pertimbangan tersebut diharapkan dapat diperoleh keuntungan yang maksimal dari sejumlah banyaknya barang yang dibawa, dengan syarat berat muatan *container* tidak melebihi kapasitas

container pengangkut, dan masing-masing *container* tersebut memiliki nilai atau *profit* yang tinggi. Inti dari permasalahan ini adalah bagaimana menentukan kombinasi barang yang akan diangkut untuk memperoleh keuntungan yang maksimal, dengan pertimbangan berat muatan tidak melebihi kapasitas alat angkut yang digunakan.

Aplikasi penyelesaian *knapsack problem* ini, dibuat dengan menggunakan algoritma genetika dan algoritma *greedy*, hasil pencarian solusi dari keduanya akan dibandingkan untuk mendapatkan hasil yang lebih optimal. Diharapkan aplikasi ini dapat membantu dalam proses seleksi barang sehingga diperoleh keuntungan atau *profit* yang maksimal.

C. Perancangan Sistem

Perancangan akan dimulai setelah tahap analisis terhadap sistem selesai dilakukan. Pada perancangan dilakukan proses menguraikan pemodelan proses alur data sistem yang akan diusulkan.

1. Perancangan Interface Aplikasi

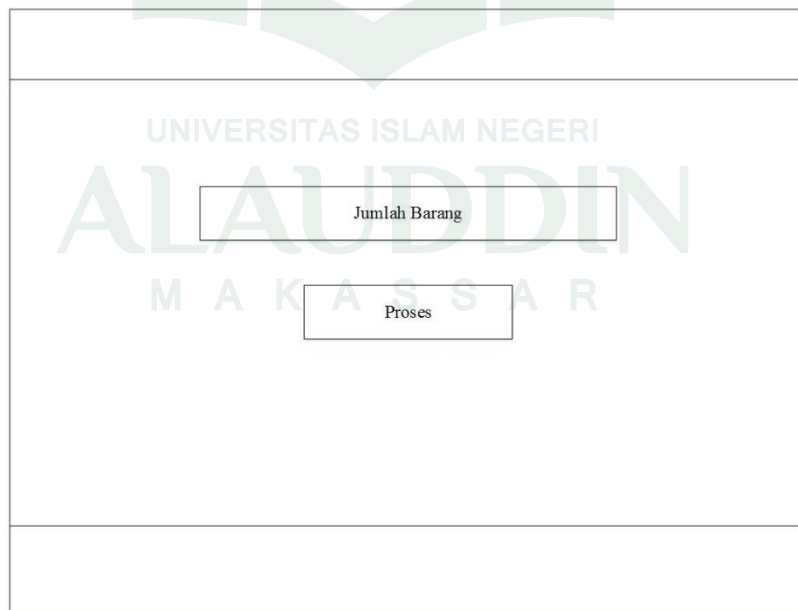
Perancangan antarmuka (*interface*) merupakan bagian penting dalam perancangan sistem, karena berhubungan dengan tampilan dan interaksi pengguna dengan sistem. Adapun perancangan antarmuka pada sistem ini adalah sebagai berikut:

a. Perancangan *Interface* Halaman Utama



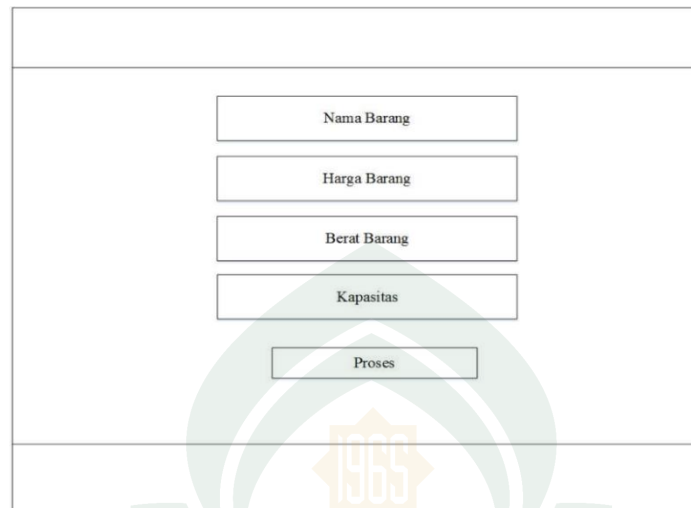
Gambar IV.2 Desain *Interface* Halaman Utama

b. Perancangan *Interface* Input Jumlah Barang



Gambar IV.3 Desain *Interface* Input Jumlah Barang

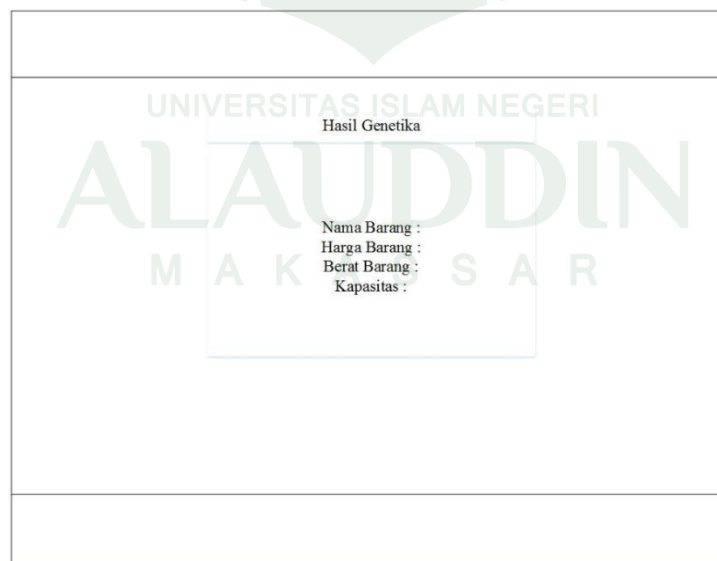
- c. Perancangan *Interface* Input Nama Barang, Harga Barang, Berat Barang dan Kapasitas



The screenshot shows a web form with four input fields stacked vertically, each with a label inside: 'Nama Barang', 'Harga Barang', 'Berat Barang', and 'Kapasitas'. Below these fields is a 'Proses' button. The form is set against a light gray background with a faint watermark of a green dome and the year '1965'.

Gambar IV.4 Desain *Interface* Input Nama Barang, Harga Barang, Berat Barang dan Kapasitas

- d. Perancangan *Interface* Hasil Genetika



The screenshot displays a web page with a header 'UNIVERSITAS ISLAM NEGERI' and a large watermark 'ALAUDDIN MAKASSAR'. The main content area is titled 'Hasil Genetika' and contains a box with the following text: 'Nama Barang :', 'Harga Barang :', 'Berat Barang :', and 'Kapasitas :'. The page has a light gray background.

Gambar IV.5 Desain *Interface* Hasil Genetika

e. Perancangan *Interface* Hasil *Greedy*

<p>Hasil Greedy</p>
<p>Nama Barang : Harga Barang : Berat Barang : Kapasitas :</p>

Gambar IV.6 Desain *Interface* Hasil *Greedy*

f. Perancangan *Interface* Hasil Kombinasi Barang Terpilih dengan Genetika

HASIL GENETIKA	
Kombinasi Barang Terpilih :	
1.	
2.	
3.	
Total Harga :	
Berat Maksimum :	
Sisa Kapasitas :	

Gambar IV.7 Desain *Interface* Hasil Kombinasi Barang Terpilih dengan Genetika

g. Perancangan *Interface* Hasil Kombinasi Barang Terpilih dengan *Greedy*

Hasil Greedy

Greedy by Weight :
Kombinasi Barang Terpilih :
1.
2.
3.

Greedy by Profit :
Kombinasi Barang Terpilih :
1.
2.
3.

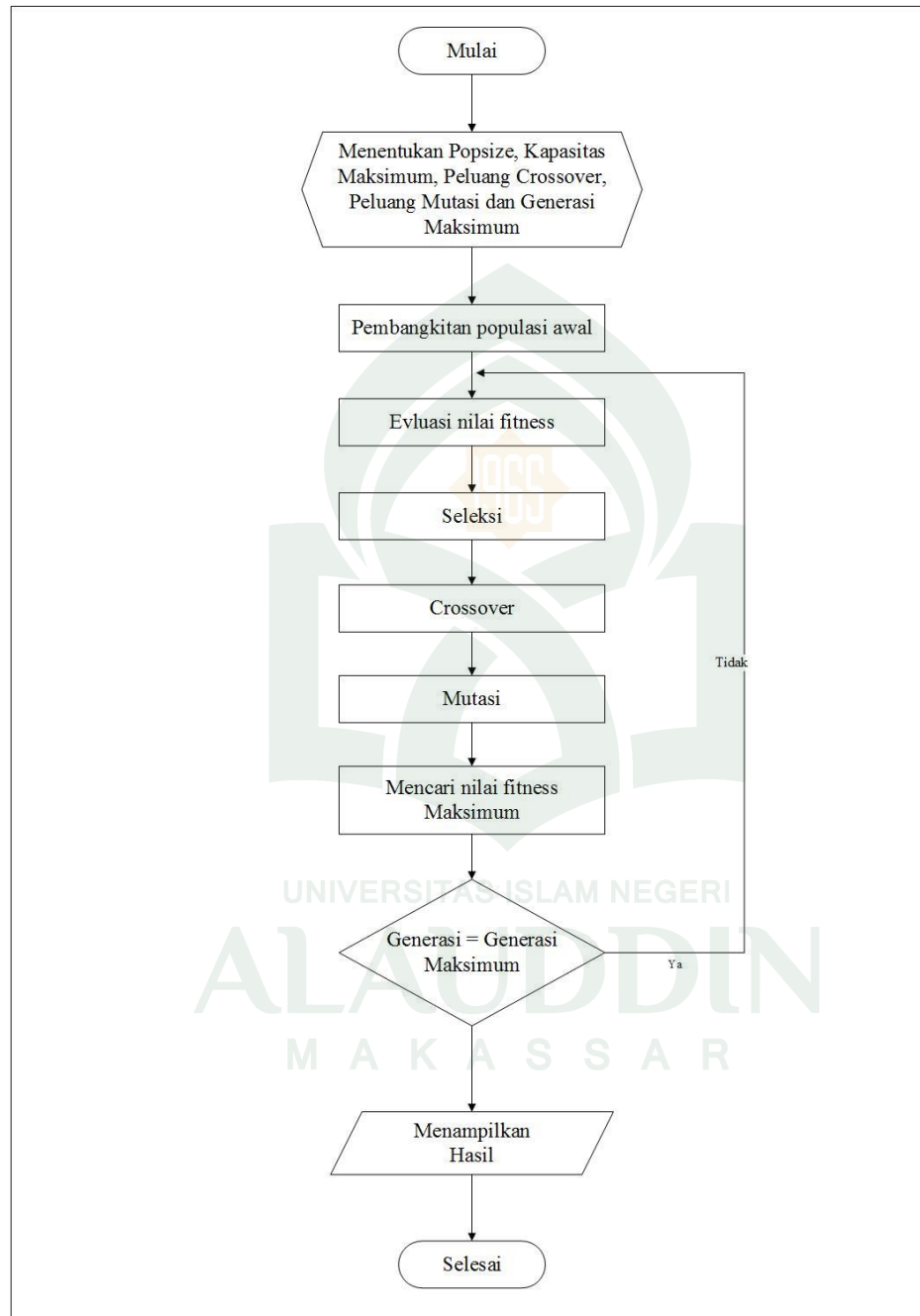
Total Harga :
Berat Maksimum :
Sisa Kapasitas :

Gambar IV.8 Desain *Interface* Hasil Kombinasi Barang Terpilih dengan *Greedy*

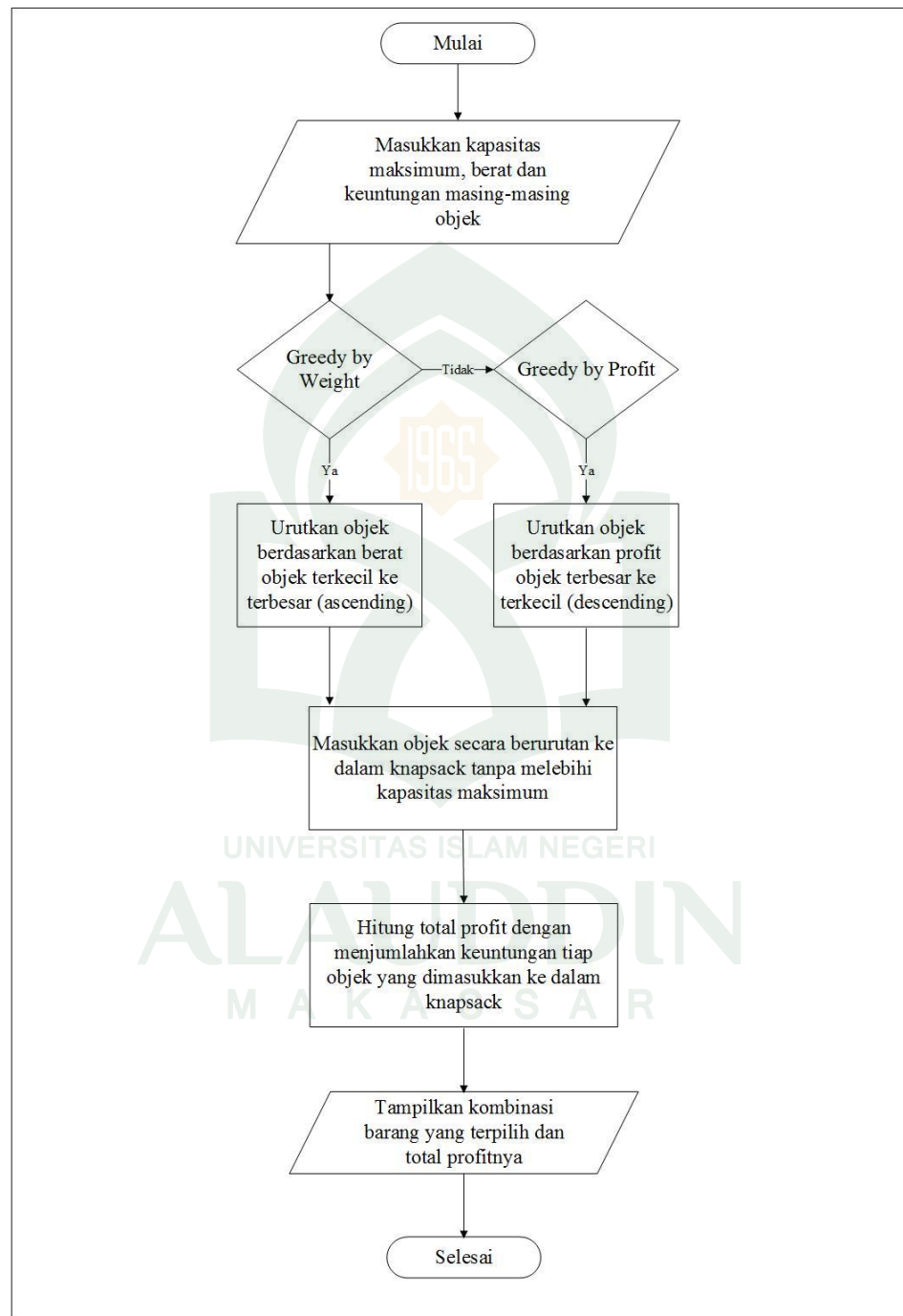
2. Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan perosedur dari suatu aplikasi.

a. *Flowchart* Algoritma Genetika



Gambar IV.9 Desain *Flowchart* Algoritma Genetika

b. Flowchart Algoritma *Greedy*Gambar IV.10 Desain *Flowchart* Algoritma *Greedy*

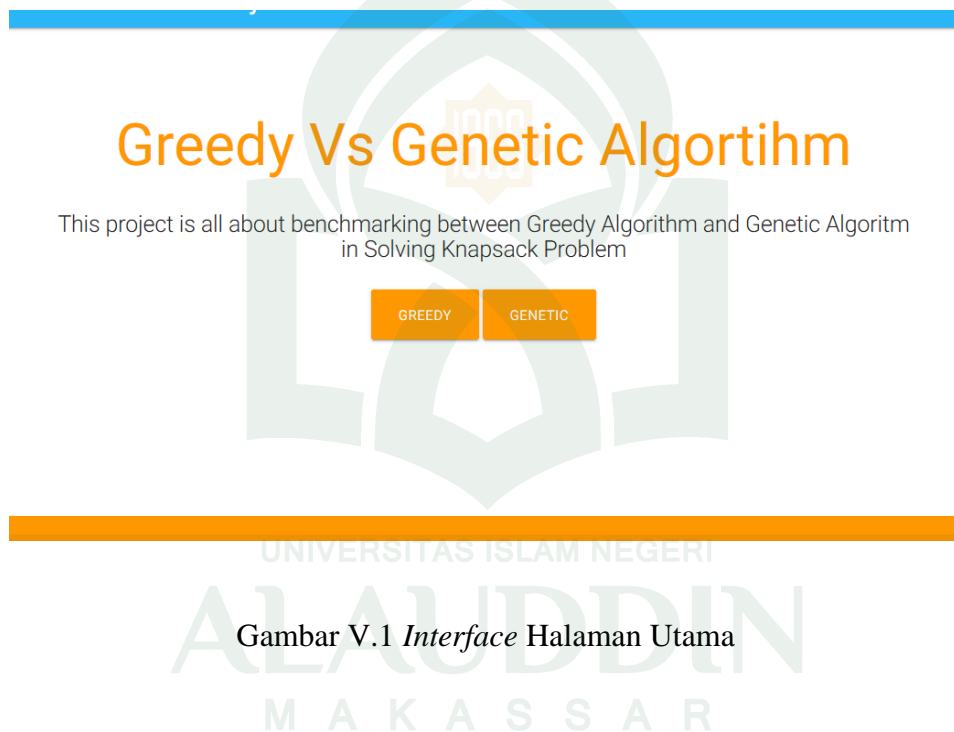
BAB V

IMPLEMENTASI DAN ANALISIS HASIL

A. Implementasi

1. *Interface* Halaman Utama

Tampilan awal aplikasi yang berisi dua menu pilihan yaitu menu pilihan *greedy* dan *genetic*.



Gambar V.1 *Interface* Halaman Utama

2. *Interface* Input Jumlah Barang dengan *Greedy Algorithm*

Antarmuka input jumlah barang dengan *greedy algorithm* muncul ketika *user* memilih menu *greedy* pada halaman utama.

Greedy Algoritma

Jumlah barang

3 | 12

PROSES

Gambar V.2 *Interface* Input Jumlah Barang

3. *Interface* Input Nama Barang, Harga Barang, Berat Barang dan Kapasitas *Greedy Algorithm*

Antarmuka input nama, harga, berat barang dan kapasitas *greedy algorithm* akan muncul sesuai dengan jumlah barang yang ditentukan pada *form* sebelumnya.

UNIVERSITAS ISLAM NEGERI

Greedy Algorithm

ALAUDDIN

MAKASSAR

peti	Nama Barang 0	
55	Berat Barang 0	
5000	Harga Barang 0	
peti	Nama Barang 1	
30	Berat Barang 1	
5000	Harga Barang 1	
peti	Nama Barang 2	
165	Berat Barang 2	
5000	Harga Barang 2	
Kapasitas		
100		

PROSES

Gambar V.3 *Interface* Input Nama, Harga, Berat Barang dan Kapasitas *Greedy Algorithm*

4. *Interface* Hasil Kombinasi Barang Terpilih dengan *Greedy Algorithm*

Antarmuka hasil kombinasi barang terpilih dengan *greedy algorithm* terdiri dari hasil pengurutan barang dengan *greedy by weight* dan *greedy by profit* beserta dengan barang yang terpilih dari hasil *greedy algorithm*.

Greedy Algorithm			
Tabel Data Barang			
Nama	Berat	Harga	Profit
peti	55	5000	275000
peti	30	5000	150000
peti	165	5000	825000
Greedy by Profit			
Nama	Berat	Harga	Profit
peti	55	5000	275000
peti	30	5000	150000
Total Profit			425000
Kapasitas			100
Sisa Kapasitas			15
Greedy by Weight			
Nama	Berat	Harga	Profit
peti	30	5000	150000
peti	55	5000	275000
Total Profit			425000
Kapasitas			100
Sisa Kapasitas			15

Gambar V.4 *Interface* Hasil Kombinasi Barang Terpilih dengan *Greedy Algorithm*

5. *Interface* Input Jumlah Barang dengan *Genetic Algorithm*

Antarmuka input jumlah barang dengan *genetic algorithm* muncul ketika *user* memilih menu *genetic* pada halaman utama.

Genetic Algoritihm

Jumlah barang

3

PROSES

Gambar V.5 *Interface* Input Jumlah Barang dengan *Genetic Algorithm*

6. *Interface* Input Nama Barang, Harga Barang, Berat Barang dan Kapasitas dengan *Genetic Algorithm*

Antarmuka input nama barang, harga barang, berat barang dan kapasitas dengan *genetic algorithm* akan muncul sesuai dengan jumlah barang yang ditentukan pada *form* sebelumnya.

Genetic Algoritihm

peti	Nama Barang 0
55	Harga Barang 0
5000	Berat Barang 0
peti	Nama Barang 1
30	Harga Barang 1
5000	Berat Barang 1
peti	Nama Barang 2
155	Harga Barang 2
5000	Berat Barang 2
Kapasitas	
100	

PROSES

Gambar V.6 *Interface* Input Nama Barang, Harga Barang, Berat Barang dan Kapasitas dengan *Genetic Algorithm*

7. *Interface* Hasil Kombinasi Barang Terpilih dengan *Genetic Algorithm*

Antarmuka hasil kombinasi barang terpilih dengan *genetic algorithm* terdiri dari hasil kromosom dengan nilai *fitness* terbaik.

Final Project

Genetic Algorithm

Tabel Data Barang			
Nama	Berat	Harga	Profit
peti	55	Rp. 5.000,00	Rp. 275.000,00
peti	30	Rp. 5.000,00	Rp. 150.000,00
peti	165	Rp. 5.000,00	Rp. 825.000,00

Yang Terbaik			
110 dengan Fitness : 10000			
Nama	Berat	Harga	Profit
peti	55	Rp. 5000,00	Rp. 825.000,00
peti	30	Rp. 5000,00	Rp. 825.000,00
Total Profit			425000
Kapasitas			100
Sisa Kapasitas			15

Gambar V.7 *Interface* Hasil Kombinasi Barang Terpilih dengan *Genetic Algorithm*

B. Pengujian Sistem

1. Pengujian *White Box*

Pengujian sistem merupakan tahap sebelum terakhir dalam pembangunan sistem. Pada tahap ini, sistem akan diuji coba baik itu dari segi logikan dan fungsi-fungsi agar layak untuk diimplementasikan. Adapun teknik pengujian sistem yang digunakan yaitu white box dengan menggunakan metode Cyclomatic Complexity (CC).

Dalam menguji sistem suatu bagan alir program (*flowchart*) yang didesain sebelumnya dipetakan ke dalam bentuk bagan alir control (*flowgraph*). Hal ini memudahkan untuk penentuan jumlah region, *Cyclomatic Complexity* (CC) dan *independent path*. Jika jumlah region, *Cyclomatic Complexity* (CC) dan *independent path* sama besar maka sistem dinyatakan benar, tetapi jika sebaliknya maka sistem masih memiliki kesalahan, mungkin dari segi logika maupun dari sisi lainnya.

Cyclomatic Complexity (CC) dapat dihitung dengan menggunakan rumus:

$$V(G) = E - N + 2$$

Dimana:

E = jumlah *edge* pada *flowgraph*

N = Jumlah *node* pada *flowgraph*

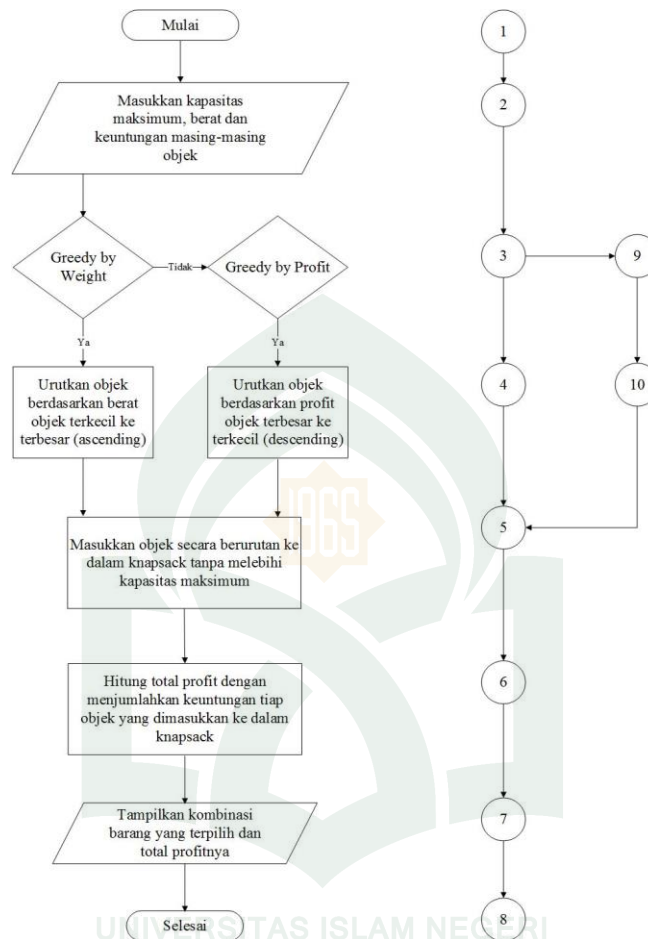
Rumusan pemetaan *flowchart* ke dalam *flowgraph* dan proses perhitungan

V(G) terhadap perangkat lunak dapat dilihat pada penjelasan berikut:

a. Prosedur Pengujian

Persiapan yang dilakukan dalam melakukan pengujian adalah sebagai berikut:

1. Menyiapkan sebuah komputer/laptop/*gadget* lainnya yang berfungsi dengan baik.
2. Membuka halaman *website* aplikasi
3. Melakukan proses pengujian
4. Mencatat hasil pengujian

a) Pengujian Algoritma *Greedy*Gambar V.8 Pengujian *whitebox* algoritma *greedy*

Diketahui :

$$E = 10 \quad N = 10 \quad R = 2$$

Penyelesaian :

$$V(G) = 10 - 10 + 2 = 2$$

Independent Path :

$$\text{Path 1} = 1-2-3-4-5-7-8$$

$$\text{Path 2} = 1-2-3-9-10-5-6-7-8$$

Tabel V. 1 Pengujian *white box* algoritma *greedy*

NODE	SOURCE
1.	<?php
2.	<pre> \$counter = \$_POST ['counter']; \$kapasitas = \$_POST ['kapasitas']; \$profit = []; \$nama = []; \$harga = []; \$berat = []; \$urutprofit = []; </pre>
3.	<pre> echo "Greedy by Weight

"; //sorting bubble sort \$weight_temp = ''; </pre>
4.	<pre> for (\$q=0; \$q < \$counter ; \$q++) { for (\$r=0; \$r < \$counter-\$q-1 ; \$r++) { if(\$berat[\$r] > \$berat[\$r+1]) { //penukaran berat \$weight_temp=\$berat[\$r]; \$berat[\$r]=\$berat[\$r+1]; \$berat[\$r+1]=\$weight_temp; //penukaran nama barang \$s=\$nama[\$r]; \$nama[\$r]=\$nama[\$r+1]; \$nama[\$r+1]=\$s; //penukaran harga barang \$t=\$harga[\$r]; \$harga[\$r]=\$harga[\$r+1]; \$harga[\$r+1]=\$t; //penukaran profit barang \$u=\$profit[\$r]; \$profit[\$r]=\$profit[\$r+1]; \$profit[\$r+1]=\$u; } } } </pre>
NODE	SOURCE
5.	<pre> \$i=0; do { \$nama[] = \$_POST['nama'.\$i]; \$berat[] = \$_POST['berat'.\$i]; \$harga[] = \$_POST['harga'.\$i]; \$i++; }while (\$i<\$counter); </pre>

	<pre> \$b=0; do{ \$profit[\$b] = \$berat[\$b] * \$harga[\$b]; \$b++; }while (\$b < \$counter); \$a=0; do { echo "Nama = \$nama[\$a]"; echo "
"; echo "Berat = \$berat[\$a]"; echo "
"; echo "Harga = \$harga[\$a]"; echo "
"; echo "profit = \$profit[\$a]"; echo "

"; \$a++; }while (\$a<\$counter); </pre>
6.	<pre> \$a=0; do { echo "Nama = \$nama[\$a]"; echo "
"; echo "Berat = \$berat[\$a]"; echo "
"; echo "Harga = \$harga[\$a]"; echo "
"; echo "profit = \$profit[\$a]"; echo "

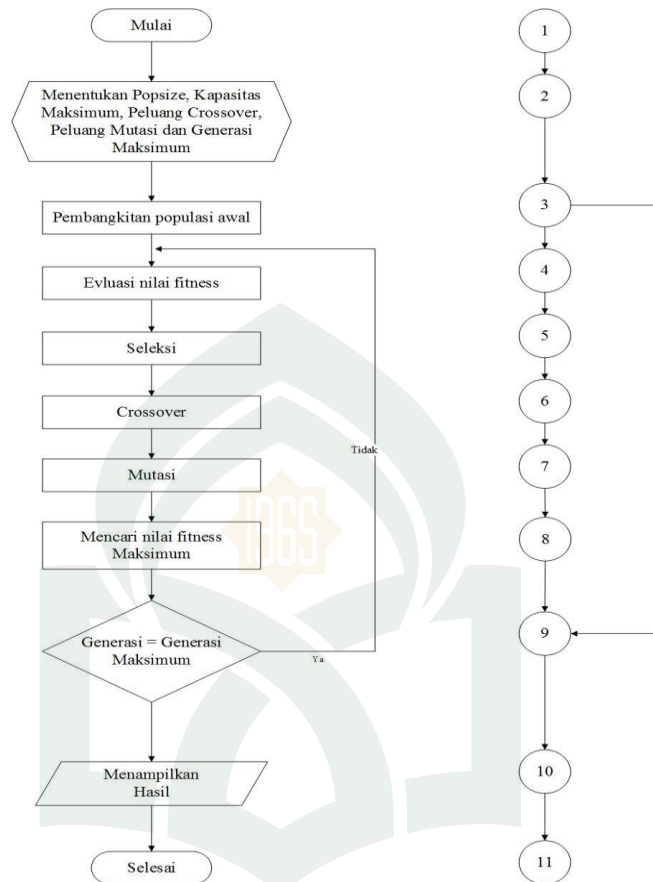
"; \$a++; }while (\$a<\$counter); </pre>
NODE	SOURCE
7.	<pre> for (\$v=0; \$v<\$counter; \$v++) { echo "\$nama[\$v] : "; echo "\$berat[\$v] : "; echo "\$harga[\$v] : "; echo "\$profit[\$v]"; echo "
"; } </pre>

	<pre> } \$kapasitasskr = \$kapasitas; echo "
"; echo "Kapasitas : \$kapasitasskr
"; for (\$g=0; \$g<\$counter; \$g++) { if (\$berat[\$g] > \$kapasitasskr) { } else { echo "\$nama[\$g] : terpilih
"; \$kapasitasskr = \$kapasitasskr - \$berat[\$g]; } } echo "sisa kapasitas = \$kapasitasskr
"; </pre>
8.	?>
9.	<pre> echo "<hr>"; echo "Greedy by Density

"; </pre>
NODE	SOURCE
10.	<pre> \$profit_temp = ''; //sorting descending bubble sort for (\$i=0; \$i < \$counter; \$i++) { for (\$j=0; \$j < \$counter-\$i-1 ; \$j++) { if(\$profit[\$j] < \$profit[\$j+1]) { //penukaran profit </pre>

	<pre>\$profit_temp=\$profit[\$j]; \$profit[\$j]=\$profit[\$j+1]; \$profit[\$j+1]=\$profit_temp; //penukaran nama barang \$k=\$nama[\$j]; \$nama[\$j]=\$nama[\$j+1]; \$nama[\$j+1]=\$k; //penukaran harga barang \$p=\$harga[\$j]; \$harga[\$j]=\$harga[\$j+1]; \$harga[\$j+1]=\$p; //penukaran berat barang \$q=\$berat[\$j]; \$berat[\$j]=\$berat[\$j+1]; \$berat[\$j+1]=\$q; } }</pre>
--	---

b) Pengujian Algoritma Genetika

Gambar V.9 Pengujian *whitebox* algoritma genetika

Diketahui :

 $E = 11$ $N = 11$ $R = 2$

Penyelesaian :

$$V(G) = 11 - 11 + 2 = 2$$

Independent Path :

Path 1 = 1-2-3-4-5-7-8-9-10-11

Path 2 = 1-2-3-9-10-11

Tabel V. 2 Pengujian *white box* algoritma genetika

NODE	SOURCE
1.	<?php
2.	<pre> \$counter = \$_POST ['counter']; //batas kontainer \$BATASKONTAINER = \$_POST ['kapasitas']; \$POPULATION = []; \$POPULATION_SIZE = 200; //pop size \$MUTATION_RATE = 0.01; //peluang mutasi \$CROSSOVER_RATE = 0.2; //peluang persilangan. bisa di definisikan sendiri. bisa by system \$DNA_SIZE = ''; \$GEN_COUNT = 1; \$TEST_COUNT = 0; \$bestGen = []; \$ITEMS = []; </pre>
3.	<pre> genInitPopulation(); function genInitPopulation(){ global \$POPULATION,\$POPULATION_SIZE; //200 for (\$i=0; \$i < \$POPULATION_SIZE ; \$i++) { \$individual = randomIndividual(); array_push(\$POPULATION, array(\$individual,fitness(\$individual))); } } </pre>

NODE	SOURCE
4.	<pre> function fitness(\$individual){ global \$ITEMS,\$GEN_COUNT,\$TEST_COUNT,\$BATASKONTAINER; \$TEST_COUNT++; \$fitness = 0; \$total_fitness = 0; \$berat = 0; \$total_berat = 0; for (\$i=0; \$i < count(\$ITEMS); \$i++) { \$fitness = \$ITEMS[\$i]- >survivalPoints * \$individual[\$i]; \$total_fitness += \$fitness; \$berat = \$ITEMS[\$i]->weight * \$individual[\$i]; \$total_berat += \$berat; } if(\$total_berat > \$BATASKONTAINER){ //jika melebihi BATASKONTAINER \$total_fitness = 0; } return \$total_fitness; } </pre>
5.	<pre> function naturalSelection(){ global \$POPULATION,\$POPULATION_SIZE,\$GEN_COUNT,\$bestGen; usort(\$POPULATION, "urutkan"); array_splice(\$POPULATION, ceil(\$POPULATION_SIZE/2)); array_push(\$bestGen, array(\$POPULATION[0][0], \$POPULATION[0][1])); // echo 'Best fit gen ' . \$GEN_COUNT . ':' . \$POPULATION[0][0] . ' (Fitness : ' . \$POPULATION[0][1] . ') ' . "\n"; // echo '
'; } </pre>

NODE	SOURCE
6.	<pre> \$crosspoint = rand(0, \$jumlahItems-1); \$ia_before_cp = substr(\$ia, 0, \$crosspoint); //\$ia_after_cp = substr(\$ia[0], \$crosspoint); //\$ib_before_cp = substr(\$ib[0], 0, \$crosspoint); \$ib_after_cp = substr(\$ib, \$crosspoint); \$child = \$ia_before_cp.\$ib_after_cp; </pre>
7.	<pre> function mutate(\$s) { global \$DNA_SIZE, \$ITEMS, \$MUTATION_RATE; \$sample = randomIndividual(); for (\$i=0; \$i<count(\$ITEMS); \$i++) { if (rand(0,100) == 100) { // \$s[\$i] = \$sample[\$i]; if(\$s[\$i] == 0){ \$s[\$i] = 0; }else{ \$s[\$i] = 1; } } } return \$s; } </pre>
8.	<pre> \$maxGene = \$bestGen[0][1]; \$bestGenMax = \$bestGen[0][0]; for (\$i=0; \$i < count(\$bestGen) ; \$i++) { if(\$maxGene < \$bestGen[\$i][1]){ \$maxGene = \$bestGen[\$i][1]; \$bestGenMax = \$bestGen[\$i][0]; } } </pre>

NODE	SOURCE
9.	<pre>function reproduction(\$ia, \$ib){ global \$DNA_SIZE, \$ITEMS; \$jumlahItems = count(\$ITEMS); \$crosspoint = rand(0, \$jumlahItems-1); \$ia_before_cp = substr(\$ia, 0, \$crosspoint); //\$ia_after_cp = substr(\$ia[0], \$crosspoint); //\$ib_before_cp = substr(\$ib[0], 0, \$crosspoint); \$ib_after_cp = substr(\$ib, \$crosspoint); \$child = \$ia_before_cp.\$ib_after_cp; \$child = mutate(\$child); return array(\$child, fitness(\$child)); }</pre>
10.	<pre>echo \$bestGenMax; echo " dengan Fitness : ".\$maxGene;</pre>
11.	?>

b. Hasil Pengujian

Tabel V.3 Rekapitulasi Hasil Pengujian Perangkat Lunak

Rekapitulasi Hasil Pengujian Perangkat Lunak					
No.	Nama Modul / Menu	CC	Region	Path	Keterangan
1.	Pengujian Algoritma <i>Greedy</i>	2	2	2	Benar
2.	Pengujian Algoritma Genetika	2	2	2	Benar

Berdasarkan hasil perhitungan region, *Cyclomatic Complexity (CC)* dan *independent path* diatas, maka dapat disimpulkan bahwa hasil pengujian sistem dengan menggunakan *white box* dengan menggunakan metode *Cyclomatic Complexity (CC)* telah benar dan tidak memiliki kesalahan baik dari segi logika maupun fungsi dan layak untuk diimplementasikan.

2. Perbandingan Algoritma *Greedy* dan Genetika

Hasil aplikasi akan diujicobakan pada data yang telah dimasukkan sebelumnya. Ada 3 indikator pengujian yang akan menjadi pembanding, yaitu solusi yang dihasilkan, maksimal profit yang dihasilkan dan kecepatan proses optimasi. Indikator ini akan diujicobakan dengan 20 barang, 30 barang dan 50 barang dengan kapasitas dan tarif yang sama. Pada algoritma genetika akan digunakan parameter berat dengan *popsiz*e 200, peluang *crossover* 0.2 dan peluang mutasi 0.01. Adapun kasus yang akan diujicobakan adalah seperti pada tabel berikut :

Tabel V. 4 Data Ujicoba Barang

No.	Nama Barang	Berat (Kg)	Harga / Tarif
1.	Peti	55	Rp.5000
2.	Peti	30	Rp.5000
3.	Peti	165	Rp.5000
4.	Kardus	25	Rp.5000
5.	Kardus	50	Rp.5000
6.	Motor	100	Rp.5000
7.	Motor	100	Rp.5000
8.	Dispenser	20	Rp.5000
9.	Jerigen	20	Rp.5000
10.	Kulkas	48	Rp.5000
11.	Kulkas	46	Rp.5000
12.	Kulkas	52	Rp.5000
13.	Lemari	62	Rp.5000
14.	Lemari	18	Rp.5000
15.	Lemari	10	Rp.5000
16.	Lemari	6	Rp.5000
17.	Lemari	57	Rp.5000
18.	Lemari	121	Rp.5000
19.	Lemari	21	Rp.5000
20.	Lemari	30	Rp.5000
21.	Karung isi underdil motor	32	Rp.5000
22.	Karung isi pakaian	21	Rp.5000

23.	Karung isi merica	13	Rp.5000
24.	Karung isi bibit jagung	25	Rp.5000
25.	Karung isi teripang	48	Rp.5000
26.	Karung isi kulit sapi	61	Rp.5000
27.	Karung isi wijen	80	Rp.5000
28.	Karung isi besi rongsokan	65	Rp.5000
29.	Karung isi spare part mobil	24	Rp.5000
30.	Peti	60	Rp.5000
31.	Peti A	35	Rp.5000
32.	Peti B	40	Rp.5000
33.	Peti C	25	Rp.5000
34.	Kardus A	30	Rp.5000
35.	Kardus B	15	Rp.5000
36.	Kardus C	18	Rp.5000
37.	Kardus D	22	Rp.5000
38.	Dispenser A	25	Rp.5000
39.	Jerigen A	10	Rp.5000
40.	Jerigen B	12	Rp.5000
41.	Kulkas A	65	Rp.5000
42.	Kulkas B	70	Rp.5000
43.	Lemari A	55	Rp.5000
44.	Lemari B	25	Rp.5000
45.	Lemari B	15	Rp.5000
46.	Lemari C	10	Rp.5000
47.	Lemari D	8	Rp.5000
48.	Lemari E	17	Rp.5000
49.	Lemari F	30	Rp.5000
50.	Lemari G	40	Rp.5000

Tabel V. 5 Perbandingan Algoritma *Greedy* dan Genetika Untuk 20 Barang dengan Kapasitas 1000 Kg

Algoritma	Indikator Pengujian		
	Solusi yang Dihasilkan	Maksimal Profit	Kecepatan Proses Optimasi
Greedy by Profit	17 jenis barang terpilih	Rp. 5.000.000 total profit yang dihasilkan	3 detik

Greedy by Weight	19 jenis barang terpilih	Rp. 4.355.000 total profit yang dihasilkan	3 detik
Genetika	19 jenis barang terpilih	Rp. 4.680.000 total profit yang dihasilkan	5 detik

Tabel V. 6 Perbandingan Algoritma *Greedy* dan Genetika Untuk 30 Barang dengan Kapasitas 1100 Kg

Algoritma	Indikator Pengujian		
	Solusi yang Dihasilkan	Maksimal Profit	Kecepatan Proses Optimasi
Greedy by Profit	15 jenis barang terpilih	Rp. 5.500.000 total profit yang dihasilkan	3 detik
Greedy by Weight	27 jenis barang terpilih	Rp. 5.395.000 total profit yang dihasilkan	3 detik
Genetika	27 jenis barang terpilih	Rp. 5.395.000 total profit yang dihasilkan	8 detik

Tabel V. 7 Perbandingan Algoritma *Greedy* dan Genetika Untuk 50 Barang dengan Kapasitas 1300 Kg

Algoritma	Indikator Pengujian		
	Solusi yang Dihasilkan	Maksimal Profit	Kecepatan Proses Optimasi
Greedy by Profit	18 jenis barang terpilih	Rp. 6500.000 total profit yang dihasilkan	5 detik
Greedy by Weight	42 jenis barang terpilih	Rp. 6.455.000 total profit yang dihasilkan	5 detik
Genetika	41 jenis barang terpilih	Rp. 6.410.000 total profit yang dihasilkan	10 detik

Dari ketiga tabel di atas dapat dilihat bahwa, algoritma *greedy by profit* selalu menghasilkan jumlah barang terpilih lebih sedikit dan memiliki total keuntungan terbesar dibanding *greedy by weight* dan genetika. Sementara algoritma *greedy by weight* menghasilkan jumlah barang terpilih dan total keuntungan yang hasilnya hampir sama dengan algoritma genetika. Hal ini dikarenakan algoritma *greedy by weight* dan algoritma genetika sama-sama menggunakan parameter berat dalam memaksimalkan keuntungannya. *Greedy by weight* memaksimalkan keuntungan dengan memasukkan objek sebanyak mungkin dengan memilih objek yang mempunyai berat paling ringan terlebih dahulu. Algoritma genetika memaksimalkan keuntungan dengan menggunakan parameter berat dengan memasukkan sebanyak-banyaknya barang akan tetapi total dari sekumpulan barang yang dipilih tidak boleh melebihi kapasitas *knapsack*. Sedangkan *greedy by profit* memaksimalkan keuntungan dengan mendahulukan barang dengan *profit* yang paling menguntungkan terlebih dahulu untuk memenuhi *space knapsack*.

Adapun dari segi kecepatan waktu yang digunakan untuk memproses optimasi barang algoritma *greedy* menggunakan waktu lebih cepat bila dibandingkan dengan algoritma genetika. Hal ini dikarenakan algoritma genetika memiliki langkah pencarian solusi yang lebih kompleks dari algoritma *greedy*.

Dari tabel di atas dapat pula dilihat bahwa algoritma genetika dengan parameter *popsiz*e 200, peluang *crossover* 0.2 dan peluang mutasi 0.01 akan menghasilkan *fitness* terbaik pada 205000 dengan jumlah kromosom terpilih atau bernilai 1 sebanyak 41.

BAB VI

PENUTUP

A. Kesimpulan

Berdasarkan pembahasan yang telah dijelaskan pada bab-bab sebelumnya, melalui tes pengujian *whitebox* dan ujicoba perbandingan algoritma *greedy* dan genetika dapat disimpulkan bahwa :

1. Algoritma genetika dan algoritma *greedy* cukup baik digunakan dalam penyelesaian *knapsack problem*. Dan dapat memberikan kemudahan kepada pihak pengelola jasa peti kemas dalam menentukan kombinasi barang yang akan diangkut sehingga dapat memaksimalkan keuntungan.
2. Algoritma genetika menghasilkan solusi optimasi kombinatorial yang mendekati solusi optimalnya dan dapat menghasilkan lebih dari satu solusi.
3. Pada algoritma genetika semakin besar jumlah populasi maka akan memberikan nilai *fitness* yang semakin tinggi, yang berarti solusi optimal yang ditemukan semakin baik. Karena peningkatan jumlah populasi akan memberikan ruang solusi yang semakin banyak.
4. Algoritma *greedy by profit* dapat memberikan profit lebih besar dibanding *greedy by weight* dan algoritma genetika.
5. Algoritma *greedy* lebih baik dalam kompleksitas waktu pencarian solusi dibanding algoritma genetika.

6. Aplikasi ini melakukan optimasi berdasarkan parameter berat dan profit dari data barang secara keseluruhan bukan dari volume barang. Sehingga kadang terjadi tidak terjangkaunya *space* yang kosong untuk penyusunan barang yang tersisa pada bagian atas. Padahal jika dilihat *space* tersebut masih dapat dimanfaatkan untuk barang yang ukurannya kecil.

B. Saran

Aplikasi penerapan algoritma genetika dan perbandingannya dengan algoritma *greedy* dalam penyelesaian *knapsack problem* ini masih jauh dari kesempurnaan. Untuk menciptakan aplikasi yang lebih baik perlu dilakukan pengembangan baik dari sisi manfaat maupun dari sisi kerja aplikasi. Berikut beberapa saran yang dapat digunakan dalam pengembangan aplikasi nantinya :

1. Aplikasi ini dapat dikembangkan dengan menambah fitur-fitur yang lebih efisien dan inovatif dalam mengimplementasikan algoritma genetika dan *greedy*.
2. Pengembangan aplikasi selanjutnya dapat menggunakan parameter volume dalam pencarian solusi sehingga *space* kosong yang tidak terjangkau dapat dimanfaatkan.
3. Aplikasi selanjutnya dapat membandingkan algoritma *heuristic* yang lain sehingga dapat diketahui algoritma mana yang paling optimal dalam menyelesaikan *knapsack problem*.

Demikian saran yang dapat penulis berikan, semoga saran tersebut dapat dijadikan sebagai bahan masukan yang dapat bermanfaat bagi penulis khususnya dan bagi pengembang pada umumnya.



Daftar Pustaka

- Agustina, Ari. Dkk. “Perbandingan Algoritma Exhaustive Search dan Algoritma Genetika untuk Memecahkan Knapsack Problem”. *Skripsi*. Malang: Fakultas Teknik Universitas Brawijaya. 2012.
- Ahmad, Basuki. Algoritma Genetika, Suatu Alternatif Penyelesaian Permasalahan Searching, Optimasi dan Machine Learning. PENTS-ITS Surabaya. 2003.
- Departemen Agama. *Al-Quran dan Terjemahannya*. Surabaya: UD. Mekar Surabaya. 2000.
- Gen M and Cheng R. *Genetic Algorithms and Engineering Design*. New York: John Wiley & Sons. 1997.
- Hasugian, Lesmardin. Pengertian Aplikasi. (Online) (<https://lesmardin1988.wordpress.com/2014/08/13/pengertian-aplikasi/#more-921>, diakses 7 Oktober 2015)
- KBBI. *Pengertian penyelesaian*. (Online) (<http://kamus.cektkp.com/penyelesaian/>, diakses 7 Oktober 2015)
- _____. *Pengertian penerapan*. (Online) (<http://kbbi.web.id/terap-2>, diakses 7 Oktober 2015)
- Kusumadewi, Sri. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu. 2003.
- Mahmudy WF. Penerapan Algoritma Genetika pada Optimasi Model Penugasan. *Jurnal Natural*. 10: 197-207.2006.
- Moelong, J. Lexi. *Metodologi Penelitian kualitatif*. Bandung: Remaja Karya, 2002.
- Nallamottu UB, Chambers TL, and Simon WE. Comparison of the Genetic Algorithm to Simulated Annealing Algorithm in Solving Transportation Location-allocation Problems With Euclidean Distances. *Proceedings of the 2002 ASEE Gulf-Southwest AnnualConference, The University of Louisiana at Lafayette*. 20-22. 2002.

Nugroho, B. *Trik dan Rahasia Membuat Aplikasi Web dengan PHP*. Yogyakarta: Gava Media. 2007.

Paryati. “Optimasi Strategi Algoritma Greedy untuk Menyelesaikan Permasalahan Knapsack 0-1”. *Skripsi*. Yogyakarta: Fakultas Teknik Universitas Pembangunan Nasional Veteran. 2009.

Prasetyowati, M. I. dan Wcaksana, Arya. “Implementasi Algoritma Dynamic Programming untuk Multiple Constraints Knapsack Problem”. *Skripsi*. Tangerang: Fakultas Teknik Universitas Multimedia Nusantara. 2013.

Pressman. *Metode Waterfall* <http://cisenaxtreme99.blogspot.co.id/2013/06/metode-waterfall-menurut-pressman.html> (9 Oktober 2015).

Purwanto, E. B. *Perancangan dan Analisis Algoritma*. Yogyakarta: Graha Ilmu. 2008.

Rosari, R. W. *PHP dan MySQL untuk Pemula*, Yogyakarta: Andi. 2008.

Shihab, M. Quraish. *Tafsir Al-Mishbah : Pesan, Kesan, Keserasian al-Qur'an*, vol 3. Jakarta : Lentera Hati. 2002.

Shihab, M. Quraish. *Tafsir Al-Mishbah : Pesan, Kesan, Keserasian al-Qur'an*, vol 15. Jakarta : Lentera Hati. 2002.

Taha HA. *Operations Research-An Introduction 6th ed*. Upper Saddle River NJ07458: Prentice Hall. 2002.

Wikipedia Website. *Algoritma Genetika*.

http://id.m.wikipedia.org/wiki/Algoritma_genetika. (7 Oktober 2015).

_____. *Aplikasi*. <http://id.m.wikipedia.org/wiki/Aplikasi>.

(7 Oktober 2015).

_____. *Knapsack Problem*.

https://en.wikipedia.org/wiki/Knapsack_problem. (9 Oktober 2015).

_____. *Optimasi*. <https://id.wikipedia.org/wiki/Optimasi>. (24 Oktober

2015).

DAFTAR RIWAYAT HIDUP



Nur Aima Ali, lahir di Watampone pada tanggal 14 Juli 1992. Anak pertama dari 8 bersaudara dari pasangan Drs. Muhammad Ali Musa, M.M. dan Dra. Aryani ini dibesarkan di keluarga yang sederhana. Pada Tahun 1999 penulis di sekolahkan di TK Pertiwi 1 Sinjai, kemudian penulis melanjutkan ke Sekolah Dasar di SD Negeri No. 23 Sinjai dan menyelesaikan SD nya pada tahun 2005. Pada Tahun 2008 Penulis menyelesaikan studi Sekolah Menengah Pertama di SMP Negeri 2 Sinjai kemudian melanjutkan ke Sekolah Menengah Atas di SMA Negeri 1 Sinjai dan menyelesaikan studi SMA pada tahun 2011. Setelah lulus, penulis melanjutkan jenjang pendidikannya di Perguruan Tinggi Universitas Islam Negeri Alauddin Makassar pada tahun 2011 jurusan Teknik Informatika dan dapat meraih gelar Sarjana Komputer (S.Kom) pada tahun 2017.